

SKROLLI

TIETOKONEKULTTUURIN ERIKOISLEHTI



KOODI

P = NP

Säteenaskellus

MicroPython ja BPI-Bit

Ruby – Japanin jalokivi

COMMODORE +
SPECTRUM 2019

Tosielämän
terminaattorit

Ajan hermolla

Minitel, C64 ja MIDI,
Philips Videopac -muistit



Mikko Heinonen
vastaava päätoimittaja

Tämäkään ei ole lehti uudistus!

En ole taikauskoinen ihminen, mutta tähän sääntöön on yksi poikkeus. Ihokarvani nousevat pystyyn aina, kun kuulen sanan ”lehti uudistus” tai ”sivustouudistus”. Molemmat kalskahtavat korvaani toimenpiteeltä, joka tehdään kun on jo hätä, ja joka vielä yleensä menee pieleen. Jokaista onnistunutta päivitystä kohti historia tuntee sekä printti- että verkkopuolelta lukemattomia tapauksia, joissa toimivasta konseptista valtaosa on nakattu roskeen vastoin parempaa tietoa ja sitten ihmetelty, minne lukijat tai kävijät katosivat. Harva sai kokea tämän niin katkeran omakohtaisesti kuin alkukesästä edesmennyt suomalaisen tietokonejournalismin pioneeri **Markku Alanen**, jonka muistoa kunnioitamme julkaisemalla hänen vuonna 2013 Skrolliin kirjoittamansa kolumnin – tällä kertaa alkuperäisessä muodossaan. Hänen tyttärensä, Skrollin alkuaikojen aktiivin **Ninnu Koskenalhon** kolumnin oheen kirjoittamaa muistojuttua ei ole toimituksessa onnistuttu lukemaan silmäkulmien kostumatta.

Siksi Skrolli ei tee lehti uudistuksia ainakaan niin kauan, kuin minä täällä huseeraan. Sen sijaan voimme kyllä hioa en-

nestään hyvää tuotetta, kuten olemme pikkuhiljaa tehneetkin aina ensimmäisestä numerosta alkaen. Se, että **Janne Sirén** löytyy tästä numerosta alkaen päätoimittajan kohdalta apinalaatikosta, ei ole uudistus vaan nykytilan toteamista. Janne on ollut täysin keskeisessä osassa lehden kokoamisessa, juttujen valinnassa ja kaikenlaisessa käytännön toimitustyössä jo vuosikaudet, joten **Tapion** päivätyön vaatiessa enemmän aikaa siirtymä oli luonnollinen.

Uudistus ei ole muuten sekään, että toinen rakas hengen tuotteeni, joukkorahoituksella synnytetty Retro Rewind -lehti, muuttuu Skrollin liitteeksi vuoden 2020 alusta. Tässä olisi nimittäin tarjolla mainio tilaisuus vesittää molempien lehtien sisältö ja saada yhteenlaskun 1+1 tulokseksi kirkkaasti alle 2. Sen sijaan ajattelimme tehdä asian oikein: Retro Rewind on kolmisenkymmentä retropelaamisen täyteistä lisä sivua Skrolliin, mutta sillä on oma toimitustiimensä ja päätoimittajanaan **Jukka O. Kauppinen**. Minä katselen edelleen näitä suurempia linjoja niin Skrollin kuin RR:nkin suhteen. 🐞



4041 0209
Painotuote
HÄMEEN KIRJAPAINO OY

SKROLLI – Tietokonekulttuurin erikoislehti

Yhteydenotot

toimitus@skrolli.fi
Ircnet: #skrolli
skrolli.fi

Vastaava päätoimittaja Päätoimittaja Toimituspäällikkö Taitto

Mikko Heinonen
Janne Sirén
Laura Pesola
Manu Pärssinen
Susanna Viljanmaa

Digipäällikkö, mediamyynti Kuvatoimittaja Talous Yhteisömestari

Toni Kuokkanen
Mitol Meerna
Anssi Kolehmainen
Mika Hyvönen

Toimitus

Jarno Niklas Alanko,
Tapio Berschewsky, Jari Jaanto,
Jukka O. Kauppinen, Ronja Koistinen

Kansikuva Sisuskuvat ▶

Kimmo Rinta-Pollari
Ronald A. Dallatorre / Wikimedia Commons
(Phalanx), Shark Robotics (Colossus), Kimmo
Rinta-Pollari, Janne Sirén

Tämän numeron avustajat

Markku Alanen, Mika Haulo,
Emma Kantanen, Tuula Keränen,
Ninnu Koskenalho, Tanja Kuokkanen,
Jarkko Lehti, Jarno Lehtinen,
Sakari Leppä, Vesa Linja-aho,
Raine Liukko, Laura Loukola, Sakari Lönn,
Lloyd Mangram, Jarkko Nääs,
Kimmo Rinta-Pollari, Mervyn J. Snafu,
Ismo Utriainen, Olli Vanajas, Joseph Watson

Julkaisija Painopaikka

Skrolli ry
Hämeen Kirjapaino oy, Tampere,
ISSN 2323-8992 (painettu)
ISSN 2323-900X (verkkójulkaisu)



- 2 **Pääkirjoitus**
- 4 **Mahdollisen rajalla**
P = NP ja muut tietotekniikan mahdollisuudet.
- 9 **Tosielämän terminaattorit**
Työn raskaan raskaat raatajat.
- 12 **Kolumni: Ninnu Koskenalho**
Markku Alasen MikroBitti.
- 14 **Kolumni: Markku Alanen**
MikroBitti in memoriam ...ja vähän Nokiakin.
- 16 **Säteenkaskellus ja etäisyyskentät**
Grafiikkamoottori pienoiskoossa.
- 22 **Kolumni: Janne Sirén**
Hyvän tietokonelehden tekijät.
- 24 **Minitel**
Ranskalainen esinetti.
- 27 **MicroPython ja BPI-Bit**
Ei MikroBitti, vaan Micro Bitti – 2010-luvun BBC Micro.
- 33 **Maailman levein demo**
Pikkujutuissa kaivattiin sisältöä pienelle 7502x1500-näytölle.
- 34 **Ruby**
Japanin jalokivi koodaajille.
- 37 **Kolumni: Ronja Koistinen**
Avaruusmarsu hurmasi lapset – koodaustarina.
- 38 **BSD-järjestelmien erot**
Se toinen avoin Unix-maailma.
- 42 **Kolumni: Vesa Linja-aho**
Jonnet ei muista Sonera-kirjaa.
- 44 **Kirjoittajan työkalupakki**
Kirjoittamisen ja viitteenhallinnan sofat.
- 49 **Ensimikro ennen mikropiirejä**
Sarjan 3. osa: näin rakennettiin tietokone 1960-luvulla.
- 52 **Videopacin massamuistit**
Tuo 70-80-luvun Philips-pelikone nykyaikaan.
- 54 **Commodore 64 ja MIDI**
Temperamenttinen ja yksilöllinen SID vahvistukseksi orkesteriisi.
- 56 **Commodore 264**
Jack Tramielin kananhuulijoutsenlaulu.
- 58 **Sirénin sisäpiiri**
Commodore/Spectrum-sisäpiiri 2019.
- 69 **Ei näin!**
Kertomatta jääneitä konsolitarinoita.
- 72 **Vanhat vampyyripelit**
JOKin pelipalstalla vanha veri janottaa.
- 74 **Scorched Earth silloin ja nyt**
Muistoja palaneilta mannuilta.
- 77 **Monipeliklassikko verkkoon**
Scorched Earth ja kumppanit internet-aikaan.
- 82 **JOKstoriaa X**
[[V]]öää.
- 84 **Lyhyet**
Skrolli 6,4 vuotta sitten ja Postipalsta.
- 85 **Skrolli-kalenteri**
Kalenterimme vuosi tulee täyteen.
- 86 **Assembly Summer -meitsie**
Skrolli-lavalla pelattiin Quake SM 2019 – ja paljon muuta.
- 88 **Skrolleri**
Skrollin demopalstalla paikataan 25 vuoden takaisia virheitä.
- 90 **Sarjakuva**
Turrikaanien yössä passiivista peliviihdettä.



Mahdollisen ja mahdottoman rajalla

Teksti: Jarno Niklas Alanko

Kuvat: Sakari Leppä, Jarno Niklas Alanko

Moderni tietokone voi vaikuttaa taikalaatikoilta, joka pystyy ihmeellisiin asioihin. On kuitenkin olemassa rajoituksia, joista mikään tietokone ei voi koskaan päästä yli.

Sannon mukaan tietotekniikassa on vain kaksi vaikeaa ongelmaa: välimuistin yhtenäisyys, asioiden nimeäminen ja off-by-one-virheet. Näiden vaikeudesta voidaan olla montaa mieltä, mutta on myös olemassa ongelmia, jotka ovat objektiivisesti vaikeita, tai jopa mahdottomia. Ensimmäinen askel kohti objektiivisuutta on poistaa ihminen kuviosta ja kysyä sen sijaan: kuinka kauan tietokoneelta kuluu ongelman ratkaisuun käyttäen parasta mahdollista algoritmia? Mutta mikä lasketaan tietokoneeksi?

Tietokone vai tilakone?

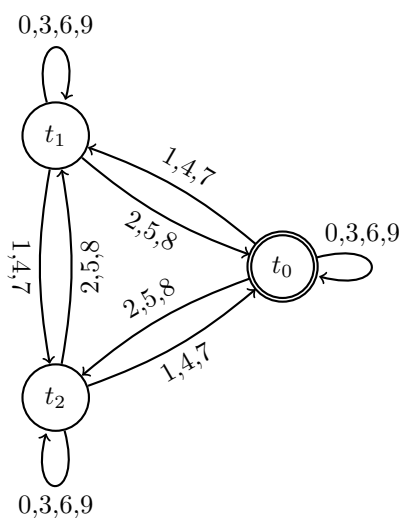
Tietokoneet pystyvät tekemään hyvin monimutkaisia asioita, mutta pohjimmiltaan kaikki perustuu vain pieneen joukkoon yksinkertaisia operaatioita. Tietokone pystyy säilömään ja hakemaan bittejä muistista ja tekemään niille yksinkertaisia laskutoimituksia. Sitä ohjaa ohjelmakoodi, jota askeltaan eteenpäin komento kerrallaan.

Muisti kuuluu olennaisena osana tietokoneeseen. Millainen olisi tietokone ilman muistia? Pieniä määriä bittejä on pakko pitää hallussa yhtä aikaa, jos halutaan laskea edes kaksi lukua yhteen. Annetaan sen verran periksi, että muistittomallakin tietokoneella saa olla pieni vakiomäärä muistia suorittimen rekistereissä, vaikka 100 tavua. Oletetaan, että ohjelmakoodi on säilötty erikseen, eikä sitä saa lukea eikä kirjoittaa uusiksi.

Tällaisella koneella pystyy tekemään jo paljon asioita. Ilman muistia tietokone on käytännössä *tilakone*. Tilakone lukee syötettä merkki kerrallaan ja vaihtaa tilaa riippuen siitä, missä tilassa se on. Nykyinen tila voidaan muistaa säilömällä sitä edustava kokonaisluku johonkin rekisteriin. Ohjelmakoodi on lista sääntöjä jokaiselle merkille, jotka

kertovat mihin tilaan siirrytään mistäkin tilasta. Esimerkiksi tilasta 5 voidaan haluta siirtyä aina tilaan 8, kun syötteestä luetaan merkki 'b'.

Tällaisella koneella voidaan esimerkiksi ratkoa, onko syötemerkkijonossa parillinen määrä b-kirjaimia. Tilakoneessa on kaksi tilaa: aina kun syötteestä luetaan b-kirjain, vaihdetaan tilaa. Jos lopullinen tila on sama kuin alkuperäinen, 'b'-kirjaimia oli parillinen määrä. Monimutkaisempiakin asioita voidaan tehdä. Tilakone voidaan ohjelmoida tunnistamaan, onko



Muistiton tilakone, joka selvittää, onko annettu luku jaollinen kolmella. Kone perustuu tietoon, että luku on jaollinen kolmella täsmälleen silloin, kun sen desimaaliesityksen numeroiden summa on jaollinen kolmella. Esimerkiksi 417 on jaollinen kolmella, koska $4+1+7=12$, ja 12 on jaollinen kolmella.

Kone lukee syötettä numero kerrallaan ja tilat esittävät tähän asti luettujen numeroiden summan jakojäännöstä kolmella. Jakojäännös on tilan nimen alaindeksissä.

Aluksi kone on tilassa t_0 ja aina kun se lukee seuraavan numeron, se siirtyy sitä nuolta pitkin, mihin on merkitty käsillä oleva numero. Esimerkiksi luettaessa luku 417, kone käy läpi tilasarjan t_1, t_2, t_0 . Jos lopuksi kone on tilassa t_0 , luettu desimaaliluku oli jaollinen kolmella.

syötemerkkijonossa jokin esiintymä annetusta säännöllistä lausekkeesta (regular expression). Näin voidaan esimerkiksi tarkistaa, onko syötejono pätevä sähköposti- tai IP-osoite.

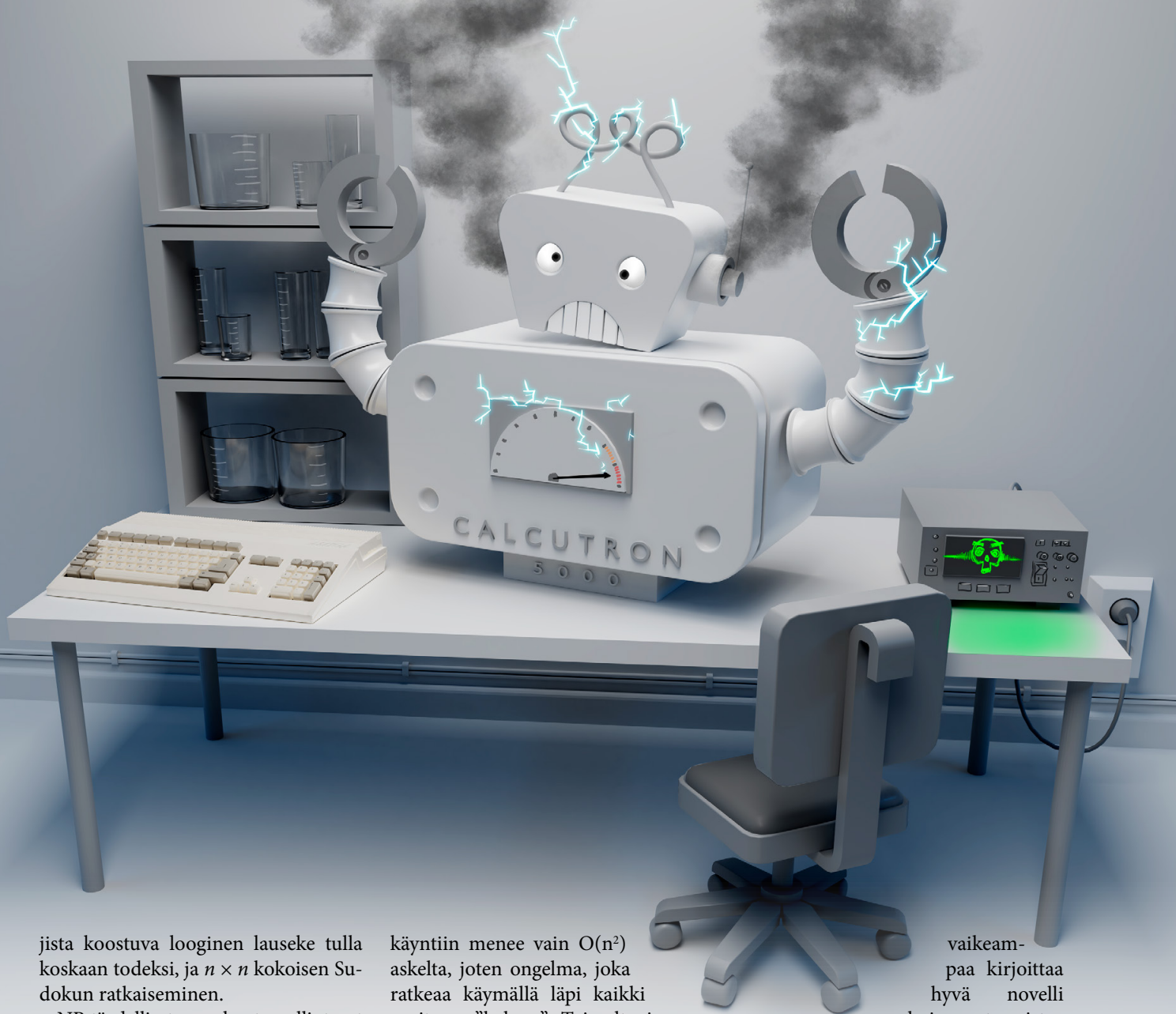
Jotkin yksinkertaisetkin asiat ovat silti mahdottomia. Otetaan esimerkki: meillä on merkkijono, jossa on plus- ja miinusmerkkejä. Kysymyksenä on, onko plus- ja miinusmerkkejä saman verran. On todistettavissa, että mikään muistiton tilakone ei selviä tästä tehtävästä. Jos annetaan tilakoneelle muisti, tehtävästä tuleekin helppo. Muisti tekee tilakoneesta tietokoneen.

Vaikeita ongelmia

Muistilla varustettu tietokone pystyy laskemaan melkein mitä vain, mutta jotkin ongelmat ovat niin hankalia, että ne ovat käytännössä mahdottomia. Otetaan esimerkiksi joukko kokonaislukuja $-56, -29, 22, 23, 34$, ja kysytään, onko olemassa osajoukkoa, jonka summa on nolla. Tässä tapauksessa vastaus on kyllä, koska $22 + 34 - 56 = 0$. Vaikka vastaus on helppo tarkistaa, tehokasta algoritmia ongelman ratkaisemiseen kaikilla syötteillä ei ole keksitty. Ensimmäiselle sellaisen keksijälle on luvassa Clayn matematiikan instituutin miljoonan dollarin palkinto ja ikuinen paikka historiankirjoissa.

Nimittäin voidaan osoittaa, että kaikki ongelmat, joiden ratkaisu on helposti tarkistettavissa, palautuvat yllä olevaan ongelmaan jollain tapaa. Ongelmia, joiden ratkaisu on helppo tarkistaa, kutsutaan NP-ongelmiksi. Lyhenne NP tulee englannin kielen sanoista *"nondeterministic polynomial time"*. Nimitys on historian painolastia siitä, että ongelmaluokka määriteltiin alun perin teoreettisten epädeterminististen Turingin koneiden avulla. Nykyaikaisempi lyhenne voisi olla vaikkapa PTC, eli *polynomial time checkable*, mutta olemme valitettavasti jumissa NP-lyhenteessä.

70-luvulla havaittiin, että monet luonnolliset NP-ongelmat ovat tarpeeksi monimutkaisia, että niillä voi mallintaa *mitä tahansa* muuta NP-ongelmaa. Näitä tietyllä tavalla riittävän rikkaita, mutta silti helposti tarkistettavissa olevia ongelmia kutsutaan NP-täydellisiksi. Esimerkiksi aiemmin mainittu summaongelma on tällainen ongelma. Muita NP-vaikeita ongelmia ovat esimerkiksi se, että voiko *and-* ja *or-*operaattoreista, suluista ja muuttu-



jista koostuva looginen lauseke tulla koskaan todeksi, ja $n \times n$ kokoisen Sudokuun ratkaiseminen.

NP-täydelliset ongelmat mallintavat myös toisiaan. Tämä tarkoittaa sitä, että kaikki NP-täydelliset ongelmat ovat teoriassa tavallaan yksi ja sama ongelma, eri muodossa esitettynä. Jos pystyt ratkaisemaan yhden, pystyt ratkaisemaan kaikki. Minkä tahansa tällaisen ongelman ratkaiseminen oikeuttaa samaan Clayn miljoonapalkintoon.

Tietojenkäsittelyteoria jakaa ongelmat karkeasti helppoihin ja vaikeisiin sen perusteella, voidaanko ne ratkoa sellaisessa määrässä askelia, jota rajoittaa jokin polynomi syötteen koosta. Polynomi koostuu siis muuttujan potenssien summasta, kuten esimerkiksi $n^2 + 4$ tai $2n^3 + 5n + 2$. Koska polynomien korkeimman asteen termi käytännössä määrittää sen kasvunopeuden, yleensä raportoidaan vain korkein aste käyttäen esimerkiksi merkintää $O(n^3)$, joka kattaa kaikki polynomit, joiden korkein aste on kolme tai alle.

Esimerkiksi jos annettuna on n alkion taulukko, kaikkien pariin läpi-

käyntiin menee vain $O(n^2)$ askelta, joten ongelma, joka ratkeaa käymällä läpi kaikki parit on ”helppo”. Toisaalta jos täytyy käydä läpi kaikki n alkion osajoukot, niin aikaa kuluu ainakin 2^n askelta, ja tällöin ongelma on ”vaikea”.

Luokan P muodostavat ne ongelmat, jotka voidaan ratkaista polynomisessa ajassa, ja luokan NP ne ongelmat, jotka voidaan tarkastaa polynomisessa ajassa. Tietojenkäsittelyteorian lipulaivaongelmaksi on muodostunut niin sanottu P vs. NP -kysymys. Se kysyy, ovatko luokat P ja NP samat. Toisin sanoen, jos ongelman ratkaisu on mahdollista tarkistaa tehokkaasti, onko silloin aina olemassa myös tehokas algoritmi ratkaisun löytämiseen, ja toisinpäin?

P vs. NP -kysymyksen filosofia

Arkitodellisuus on täynnä ongelmia, joiden ratkaisu on helpompi tarkastaa kuin keksiä tyhjästä. On esimerkiksi paljon vaikeampaa koota palapeli kuin vahvistaa, että valmiiksi koottu palapeli on koottu oikein. On myös paljon

vaikeampaa kirjoittaa hyvä novelli kuin tunnistaa hyvä novelli. Intuitiivisesti saattaa vaikuttaa siis todennäköiseltä, ettei P ole sama kuin NP.

Asian todistaminen matemaattisesti on vain osoittautunut erittäin vaikeaksi. Tietojenkäsittelytieteilijä **Scott Aaronson** on miettinyt, että todistuksen sijaan voisimme julistaa $P \neq NP$ luonnonlaiksi, joka pätee, kunnes törmäämme ensimmäiseen vastaesimerkkiin. Tämä voisi olla vaikka tietojenkäsittelyteorian ensimmäinen laki.

Jotain tämänkaltaista on käytännössä tehtykin tietojenkäsittelyteorian yhteisössä. Yleisesti otaksutaan, että $P \neq NP$. Jos yrität ratkoa ongelmaa, mutta huomaatkin, että se on yhtäpitävä jonkin NP-täydellisen ongelman kanssa, niin silloin kannattaa lopettaa taistelu luontoa vastaan. Jos onnistuisit ratkaisemaan ongelmasi, ratkaisisit samalla siinä sivussa yhden modernin tieteen tärkeimmistä avoimista kysymyksistä, jota lukemattomat viisaat ihmiset ovat jo pohtineet tuloksetta.

Vaikka P vs. NP -kysymystä ei ole onnistuttu ratkaisemaan, on asian tiimoilla silti tapahtunut jotain edistystä. On osoitettu, että tietyntyyppiset todistusstrategiat eivät tule koskaan toimimaan. Vaikuttaa hedelmällisimmältä lähestyä ongelmien vaikeuden luokittelua aivan toisia polkuja pitkin.

Informaatio algoritmin pullonkaulana

P vs. NP -kysymys on hyvin karkea jaottelu: esimerkiksi aikavaativuudet $O(n)$ ja $O(n^3)$ ovat molemmat luokassa P, vaikka käytännössä ero on valtava. Ongelmien vaativuuksien erottelu luokan P sisällä vaatii täsmällisempiä työkaluja.

Yksi käyttökelpoinen työkalu analyysiin on algoritmin käyttämä informaatio. Otetaan esimerkkinä taulukon järjestäminen. Kaikkien hyvien yleiskäyttöisten järjestysalgoritmien, kuten lomitusjärjestämisen (merge sort) ja kekojärjestämisen (heap sort), aikavaativuus on $O(n \log n)$ operaatiota. Pystyttäisiinkö parempaan? Olisiko $O(n)$ ratkaisu mahdollinen?

Vastaus on kielteinen, jos asetetaan seuraava rajoitus: algoritmi ei saa tie-

tää taulukon elementtejä, mutta saa kysellä kahdesta elementistä, kumpi näistä on suurempi. Jokaisessa iteraatiossa algoritmi saa siis vertailla yhtä paria ja sen jälkeen tehdä mitä vain muuta, kuten vaikka siirrellä elementtejä ympäriinsä ja tallentaa muistiin tietoa.

Taulukon olennainen informaatio järjestämisen kannalta on sen alkioiden keskinäinen järjestys. Erilaisia järjestyksiä on $n!$ (eli luvun n kertoma, esimerkiksi $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$), joten olennaisesti erilaisia syötteitä on $n!$. Jokaista näistä syötteistä tulee käsitellä eri tavalla, koska jos kaksi eri lähtöjärjestystä järjestetään samalla tavalla, vähintään toinen menee väistämättä väärin.

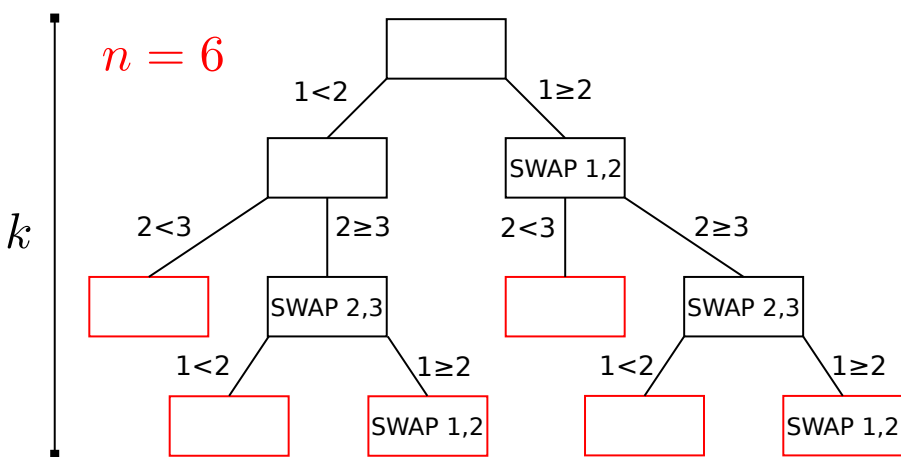
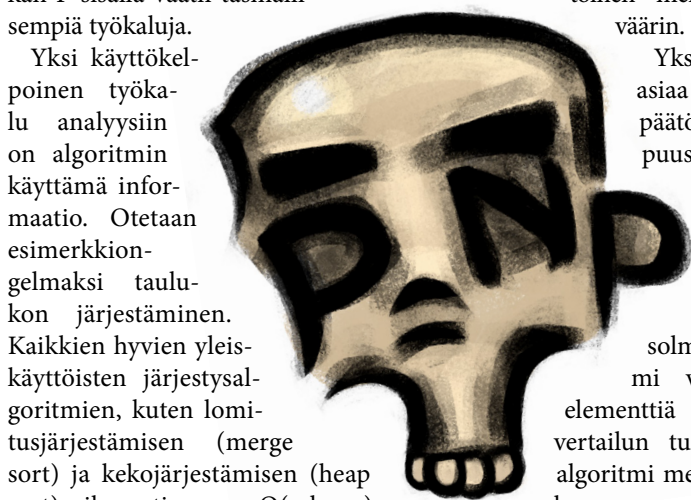
Yksi tapa katsoa asiaa on kuvitella päätöspuu. Polut puussa kuvaavat algoritmin suorituspolkuja. Suoritus lähtee juuresta, ja jokaisessa solmussa algoritmi vertailee kahta elementtiä keskenään. Jos vertailun tulos on "kyllä", algoritmi menee vasempaan haaraan, muutoin oikeaan haaraan. Lopussa erillisiä lehtiä täytyy olla vähintään $n!$, koska jokainen erilainen lähtöjärjestys tulee käsitellä eri tavalla. Tästä seuraa, että päätöspuun syvyys eli algoritmin suo-

ritusaika on vähintään luokkaa $n \log n$ (kaavio 1).

Toinen tapa lähestyä asiaa on pitää kirjaa algoritmin saamasta informaatiosta biteissä. Vertailun $x < y$ tulos on joko kyllä tai ei, joten siitä saadaan yksi bitti informaatiota taulukosta. Algoritmin on saatava vähintään sen verran informaatiota, että se pystyy yksilöimään taulukon alkuperäisen järjestyksen. Erilaisia järjestyksiä on $n!$, joka vastaa luokkaa $n \log n$ bittiä informaatiota, joten tarvitaan luokkaa $n \log n$ vertailua. Siispä taulukko on yleisessä tapauksessa mahdoton järjestää nopeammin kuin $O(n \log n)$ ajassa.

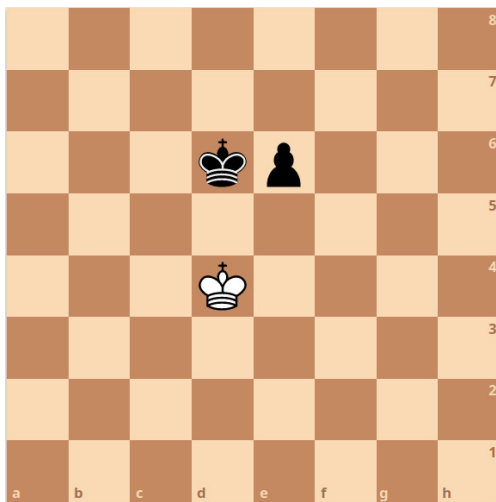
Tällaisia argumentteja kutsutaan *informaatioteoreettiksi alarajoiksi*. Jos algoritmi saa n bittiä informaatiota syötteestä, sillä voi olla korkeintaan 2^n eri suorituspolkua (olettaen, että algoritmi ei käytä satunnaisuutta). Jos ongelma vaatii yli 2^n erilaista suorituspolkua, niin algoritmi ei voi toimia oikein kaikilla syötteillä.

Kaikki järjestysalgoritmit eivät kuitenkaan sovellu tähän malliin. Esimerkiksi jos tiedetään, että taulukon alkiot ovat lukuja väliltä yhdestä kymmeneen, voidaan vain laskea yhdellä läpikäynnillä, kuinka monta mitään lukua on, ja sitten tulostaa oikea määrä kaikkia lukuja peräjälkeen. Tähän menee vain $O(n)$ aikaa, mikä vaikuttaisi olevan ristiriidassa edellisen todistuksen kanssa. Ristiriitaa ei kuitenkaan ole, koska oletus, että kaikki taulukosta saatava tieto tulee alkioiden parittaisen vertailujen kautta, ei enää päde.



$$\begin{aligned}
 k &\geq \log_2 n! \\
 &= \sum_{i=1}^n \log_2(i) \\
 &\geq \sum_{i=n/2}^n \log_2(i) \\
 &\geq \frac{n}{2} \log_2(n/2) \\
 &= \frac{n}{2} (\log_2(n) - 1) \\
 &\in \Omega(n \log n)
 \end{aligned}$$

Kaavio 1: Päätöspuu kolmen elementin taulukon järjestämiselle. Jos kaaressa lukee esimerkiksi $1 < 2$, niin silloin on saatu tieto, että $v[1] < v[2]$, missä v on järjestettävä taulukko. Jos kaksi elementtiä on väärinpäin, ne vaihdetaan päittäin. Lehtien määrä on $n!$, ja puun syvyys on k . Koska joka tasolla solmujen määrä korkeintaan tuplaantuu, ja lehtiä tarvitaan vähintään $n!$, täytyy olla $2^k \geq n!$, eli $k \geq \log_2 n!$. Tästä seuraa, että puun korkeus eli algoritmin suoritus aika on vähintään luokkaa $n \log n$.



Siirto	Numero	Koodi
e5	0	000
Kc6	1	001
Kc7	2	010
Kd7	3	011
Ke7	4	100

Kaavio 2: Kaikki mustan viisi mahdollista siirtoa on järjestetty aakkosjärjestykseen, ja jokaiselle on annettu binäärikoodi järjestyksen mukaan.

Pakkauksen rajat

Tietokoneen yksi tärkeä ominaisuus on tiedon pitkäaikainen säilytys. Kun tietomassa kasvaa suureksi, on yleensä taloudellisesti kannattavaa ajaa se jonkin pakkausalgoritmien läpi. Tiedon pakkautuvuudelle on kuitenkin olemassa erilaisia teoreettisia rajoja.

Kuvitellaan, että pyöritämme maailman suurinta online-shakkipalvelinta. Palvelimella pelataan joka päivä satojatuhansia pelejä, ja haluamme säilöä kaikki pelit tietokantaan. Koska pelejä on paljon ja tilaa rajallisesti, haluamme minimoida tilankäytön. Kuinka monta bittiä tarvitaan yhden pelin tallentamiseen?

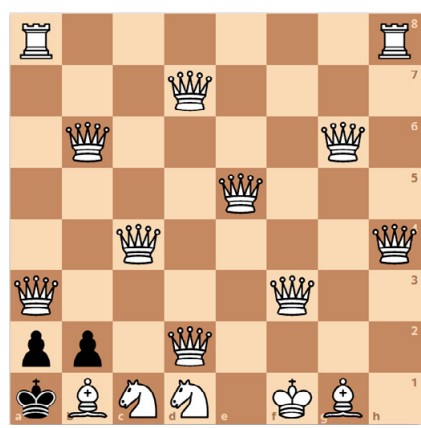
Ensimmäinen yritys voisi olla esittää siirrot pareina peliruutuja, jotka kertovat siirron lähtöruudun ja kohderuudun. Peliruutu voidaan esittää x- ja y-koordinaateilla. Esimerkiksi siirto ((2,3),(4,2)) tarkoittaisi, että nappula ruudusta (2,3) siirtyy ruutuun (4,2). Koska shakkilaudan leveys on 8 ruutua, yksi koordinaatitiluku mahtuu kolmeen bittiin, ja kokonainen siirto $4 \times 3 = 12$ bittiin.

Parannettavaa on kuitenkin vielä, koska tämä koodaus esittää myös siirtoja, jotka eivät ole mahdollisia pelin säännöissä. Esimerkiksi torni ei voi liikkua vinottain eikä toisten nappuloiden lävitse. Säästeliäämpi koodaus esittää vain sallitut siirrot. Tämä voidaan toteuttaa esimerkiksi listaamalla mahdolliset lautatilanteet siirron jälkeen, järjestämällä ne jollain yhdenmukaisella tavalla ja koodaamalla siirron järjestysluku sallittujen siirtojen listassa (kaavio 2). Ylärajaa sallittujen siirtojen määrälle lautatilanteessa ei tiedetä, mutta maailmanennätys on

Nenad Petrovicin keksimä 218 siirron keinotekoinen tilanne vuodelta 1964 (kaavio 3). On siis melko turvallista olettaa, että yli 256 laillisen siirron tilannetta ei tule vastaan, joten siirron tallentamiseen riittää kahdeksan bittiä.

Pakkausta pystytään edelleen parantamaan, jos otetaan mukaan lisää tietämystä pakattavasta datasta. Siirrot eivät ole satunnaisia, vaan tietyt siirrot ovat todennäköisempiä kuin toiset. Pelaajat harvoin siirtävät yksinäistä reunasotilasta, jos muualla laudalla on tarjolla ilmeinen shakkimatti.

Tehokas pakkausalgoritmi mallintaa seuraavan siirron todennäköisyysjakauman ja hyödyntää tätä koodauksessa. Näin päästään lähelle eräänlaista informaatioalarajaa: datan entropiaa. Informaatioteoriassa entropia mittaa jakauman tasaisuutta. Mitä epätasaisempi seuraavan siirron jakauma on, sitä pienempään tilaan seuraava siirto voidaan pakata, koska todennäköi-



Kaavio 3: Shakin sääntöjen puitteissa saatettava pelitilanne, jossa valkealla on 218 mahdollista siirtoa.

simmat siirrot voidaan koodata pienemmällä määrällä bittejä esimerkiksi Huffman-koodauksella.

Mutta edelleen voidaan parantaa, jos otetaan siirrot siirtopareina ja mallinnetaan jokaisen siirtoparin todennäköisyys. Tällöin pakkaus mahtuu siirtoparien entropian kokoiseen tilaan, joka voi olla pienempi kuin yksittäisten siirtojen entropia. Samalla logiikalla voidaan pakata siirtokolmikkoja, siirtonelikkoja ja niin edelleen. Ajatuksen looginen päätepiste on se, että koodataan jokainen peli siten, että pelin binääriesityksen pituus riippuu vain koko pelin todennäköisyydestä. Mitä todennäköisempi peli, sitä lyhyempi koodi. Tätä ajatusta ei voi enää venyttää pitemmälle – päästiinkö nyt ultimaattisen pakkauksen äärelle?

Ei välttämättä. Tämä malli olettaa, että pelien todennäköisyydet pysyvät aina samana. Tämä ei kuitenkaan pidä paikkaansa oikeasti, koska esimerkiksi avaukset saattavat tulla muotiin ja poistua muodista. Mahdottomasta voi tulla mahdollista, kun laajentaa mallia. Mitä enemmän datan rakenteesta tiedetään, sitä enemmän sitä pystytään pakkaamaan.

Ultimaattinen pakkausmenetelmä olisi sellainen, joka ottaa tietokannan pelejä ja tulostaa tietokoneohjelman, joka tulostaa tietokannan pelit. Ohjelma voidaan esittää vaikka C-koodina. Äkkiseltään voisi ajatella, että ohjelmointikielen valinnalla on väliä, mutta itse asiassa se vaikuttaa vain vähän. Jos pienempi ohjelma olisi mahdollista tehdä jollain muulla ohjelmointikielillä, vaikka Pythonilla, C-koodiin voitaisiin rakentaa Python-tulkki ja mukaan liittää Python-koodi. Kun pakattavan aineiston koko kasvaa suureksi, Python-tulkin viemä tila pienenee merkityksettömän pieneksi koko aineistoon verrattuna.

Valitettavasti ei ole olemassa algoritmia, joka tulostaa kaikissa tapauksissa pienimmän mahdollisen tietokoneohjelman annetulle tietokannalle. Eikä koskaan voikaan olla olemassa. Pienimmän mahdollisen ohjelman löytävä algoritmi johtaisi matemaattiseen ristiriitaan. Mahdottomuustodistus ei vaadi mitään matemaattisia esitietoja, ja se löytyy oikeasta laatikosta.

Kaikkien ongelmien äiti

Lyhyt kierroksemme mahdottomuustuloksien teoreettisessa maailmassa

on tulossa päätökseen, mutta loppuhiipennuksena esitän vielä kaikkien vaikeiden ongelmien äidin. Tämä ongelma on niin vaikea, että lähes kaikki tietokoneella ratkeavat ongelmat voidaan nähdä tämän erikoistapauksina. Ongelman nimi on pysähtymisongelma, ja se kuuluu seuraavasti: tietokoneohjelman koodista halutaan tietää pysähtyykö se, vai jääkö se ikuisen silmukkaan.

Otetaan nyt mikä tahansa ongelma, joka on laskettavissa tietokoneella. Esimerkiksi jutun alussa esitelty summaongelma, eli onko annetussa lukujoukossa osajoukkoa, jonka summa on nolla. Tämä voidaan palauttaa pysähtymisongelmaan seuraavasti: tehdään ohjelma, joka kokeilee kaikkia mahdollisia osajoukkoja. Jos sopiva osajoukko löytyy, ohjelma pysähtyy. Jos mikään osajoukoista ei kelpaa, ohjelma ohjataan ikuisen silmukkaan.

Nyt jos meillä olisi algoritmi, joka ratkaisisi pysähtymisongelman, siltä voitaisiin kysyä, pysähtyykö yllä rakentamamme summaongelma-algoritmi annetulla syötteellä. Koska se pysähtyy vain, jos on olemassa nollaan summaava osajoukko, hypoteettinen pysähtymisongelma-algoritmi ratkaisee myös summaongelman.

Summaongelman tilalla olisi voinut olla mikä tahansa ongelma, jonka ratkaisu on mahdollista tarkistaa, ja jonka ratkaisut on mahdollista listata. Ratkaisuehdokkaita saa olla jopa ääretön määrä. Esimerkiksi kuuluisa Fermat'n suuri lause sanoo, että yhtälöllä $x^n + y^n = z^n$ ei ole ratkaisua positiivisilla kokonaisluvuilla, kun $n \geq 3$. Jos meillä olisi algoritmi pysähtymisongelmaan, voisimme todistaa Fermat'n suuren lauseen kysymällä: pysähtyykö sellainen ohjelma, joka etsii vastaesimerkkiä ja pysähtyy sen löytäessään?

Koska pysähtymisongelma on niin ilmaisuvoimainen, sen ratkaisemisen täytyy olla äärimmäisen vaikeaa. Osoittautuukin, että ratkaiseminen on mahdotonta tietokoneella. Tämä todistetaan rakentamalla juonikas ohjelma, joka käyttää pysähtymisongelman ratkaisua alirutiinina, ja toimii päinvastoin, kuin alirutiini ennustaa sen toimivan. Tuloksena on matemaattinen ristiriita, koska rakennettu ohjelma pysähtyy täsmälleen silloin, kun se ei pysähdy. Näin ollen ratkaisua pysähtymisongelmaan ei tulla koskaan näkemään. 🦋



AIVOJUMPPAA

Miksi Kolmogorov-pakkaus on mahdotonta?

Kolmogorov-esitys merkkijonolle x on pienin mahdollinen ohjelma, joka tulostaa x . Unelmien pakkausalgoritmi olisi ohjelma, joka osaisi tulostaa lyhyimmän mahdollisen ohjelman mille vain merkkijonolle x .

Oletetaan, että on olemassa tällainen unelmien pakkausohjelma K . Olkoon K :n ohjelmakoodin koko vaikkapa megatavu konkreettisuuden vuoksi. Ohjelman K olemassaolo johtaa kuitenkin ristiriitaan seuraavalla tavalla:

Rakennetaan uusi ohjelma M , joka kokeilee merkkijonoja systemaattisesti, kunnes löytää merkkijonon z , jota ei voi generoida alle gigatavun kokoisella ohjelmalla, ja sitten tulostaa sen. M on helppo ohjelmoida alle gigatavulla koodia kokeillen merkkijonoja yksitellen ja käyttämällä ohjelmaa K alirutiinina kertomaan, kuinka suuren ohjelman mikäkin merkkijono vaatii. Mutta nyt meillä on alle gigatavun ohjelma M , joka tulostaa merkkijonon z , minkä piti olla mahdotonta alle gigatavussa. Ristiriita!

Oli ohjelman K koko mikä tahansa, voidaan aina johtaa vastaava ristiriita ottamalla tarpeeksi pitkä merkkijono z todistukseen. Ainoaksi mahdollisuudeksi jää, että ohjelmaa K ei voi olla olemassa.

TOSIELÄMÄN TERMINAATTORIT

Itävaltalais-amerikkalainen kehonrakentajanäyttelijä Arnold Schwarzenegger antoi tapajaroboteille kasvot vuoden 1984 hittielokuvassa *Terminator – tuhoaja*. Tosielämän terminaattorit saivat kuitenkin alkunsa paljon aikaisemmin.

Teksti: Janne Sirén

Kuvat: Wikimedia Commons (War Office / Horton, US Army, US Air Force / Leslie Pratt, DARPA), NASA / Anne McClain, Kimmo Rinta-Pollari



Iso-Britannia, 6. kesäkuuta, 1941. Tuimasta ilmeestä mustavalkoisessa valokuvassa ei voi erehtyä, vaikka kuuluisa sikari puuttuu suupielestä ja takki saa tuoreen pääministerin näyttämään tavallista hoikemmalta. Valokuvassa poseeraa ilmiselvästi **Winston Churchill** esikuntineen.

Churchillin päässä on toinen hänen hattusuosikeistaan: Ei se *Homburg*, jollaista myös **Al Pacino** käytti *Kummi-sedässä* ja jollaisen eräs brittijuontaja antoi presidentti **Donald Trumpille** televisiossa, vaan korkeampi *Cambridge Bowler*, kuuluisalta hatturilta Lockilta. Samanlaista käytti Bondpahis **Oddjob** lentävänä aseena. Churchill kutsui hattuaan Bowkeriksi.

Bowkerin takaa pilkistää kuitenkin kuvan todellinen pahis: L-5894. Lennokki. Drooni. Drone.

Ei tullut Robocopia

Populaarikulttuurissa lähitulevaisuuden tuhoajarobotit näyttäytyvät usein koneen ja ihmisen yhdistelminä – kenties siksi, että yleisön on helpompi samaistua inhimillisiin hahmoihin. Ne on tietenkin ollut myös helpompi toteuttaa. *Terminator – tuhoaja* -elokuvan päähahmo oli ihmiseltä näyttänyt robotti, jota on joskus virheellisesti kutsuttu myös kyborgiksi. Terminaattori oli kuitenkin fiktiivinen biorobotti, ei kyborgi, koska sen pohjana ei ollut biologinen luovuttaja.

Toinen terminaattorin aikalainen, mekaaninen mellestäjä nimeltään *RoboCop* (1987), sen sijaan oli kyborgi – ihmisen ja robotin yhdistel-

mä. Kolmannesta ihmisen ja robotin yhdistelmästä olemme kirjoittaneet Skrollissakin: mechat ovat ihmismäisiä jättirobotteja, joiden ohjaksissa hääriilee ihminen (Skrolli 2018.2). Kun elokuvissa tai videopeleissä robotti tekee tuhojaan, se onkin usein biorobotti, kyborgi tai mecha.

Todellisuudessa näistä visioista ollaan kuitenkin kaukana. Lähimmäksi tosielämän biorobottia tai kyborgia taidettiin päästä tiedekeskus Heureka keväällä, kun neurooverkkosyntetisaattori cellF pörisi omiaan pienen biologisen solukon pohjalta (ks. Skrolli 2019.2). Mechoja on oikeastikin, mutta vähäisessä määrin, koska on epäselvää, kuinka hyödyllistä ihmistä ja robottia on ylipäättänsä yhdistää fyysiseksi kokonaisuudeksi. Mechateknologian suurin lupaus saattaa olla vammaanhoidossa.

Tosielämän terminaattorit ovat paljon tylsempiä ja tyhmempiä, mutta silti tehokkaita tuottamaan niitä vammoja.

Kuhnureita vai lennokkeja

Etenkin valokuvauskäytössä yleistyneiden nelikoptereiden myötä Suomessa on keskusteltu siitä, miksi lentäviä robotteja tulisi kutsua. Viime vuoden lopulla Helsingin Sanomat linjasi kutsuvansa niitä lennokeiksi. Hesari kulki tässä jossain määrin vastavirtaan, sillä muut mediat ja Suomen kielitoimisto olivat tarjonneet nelikoptereiden nimeksi englanninkielisestä kantasanaa drone väännettyä vierassanaa drooni.

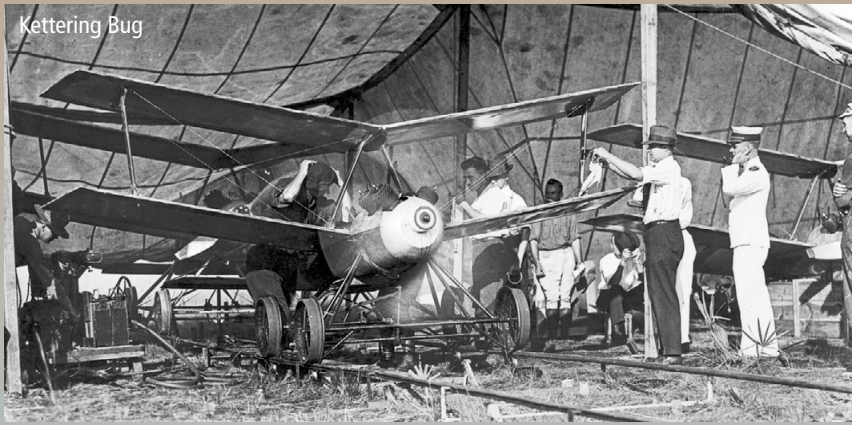
Tämä keskustelu on ollut paikoin pinnallista ja historiatonta, mutta se johtaa meidät tosielämän terminaattori-

toreiden jäljille – siihen kesäkuuseen perjantaihin Norfolkkin rannikolla 1941. Winston Churchill oli nimittäin saapunut seuraamaan valokuvassa näkyvän, de Havilland DH.82 Tiger Mothilta näyttävän kaksitasolentokoneen lentoa. Lennosta teki erityisen se, että kone oli muutettu radio-ohjatuksi tykistömaaliksi. Konevariantille, jota valmistettiin satoja, oli annettu nimi: Queen Bee.

Queen Bee ei tosin ollut enää mikään uusi keksintö vuonna 1941. Ensimmäiset kauko-ohjatut robottimaalit juontavat juurensa 1910-luvulle. Yksi varhaisista oli **Archibald Montgomery Low'n** kehittämään radiotekniikkaan perustunut Ruston Proctor Aerial Target (1916). Queen Been nimi – mehiläiskuningatar – on kuitenkin tarinan kannalta merkityksellinen. Se kun viittaa mehiläispesään, josta löytyy myös kuhnureita.

Kuhnurit (engl. *drone*) ovat mehiläispesän turhakkeita, jotka lentävät tyhminä parvina ainoastaan paritelakseen kuningattaren kanssa. Niillä ei ole piikkiä, eikä mitään muuta tehtävää. Tarkalleen ottaen nimensä mukaisesti Queen Bee houkutteli puoleensa kuhnureita, mutta etenkin yleisön suussa drone-nimitys siirtyi tyhminä lentäviin tykistömaaleihin. Jotkut lähteet viittaavat myös maalien tasaisen surisevaan ääneen (engl. *droning*) sanan takana.

Yksi syy drone-nimityksen yleistymiselle oli epäilemättä myös se, että englannin kielestä puuttui luonteva sana ilmaa raskaammalle miehittämät-



tömälle lentävälle esineelle. *Unmanned Aerial Vehicle* (UAV) kun ei suoranaisesti kierähdä kielen päältä. Suomessa sana sen sijaan oli: meillä lentäviä tykistömaaleja kutsuttiin lennokeiksi.

Drooni L-5894:ää ei ammuttu alas tuona päivänä, se tapahtui vasta paria

viikkoa myöhemmin Weybournessa. Sotilaslennoykkien historia ulottuu kuitenkin vielä 1900-luvun alkupuoliskoa pidemmälle – sekä ajassa että käyttötarkoituksessa.

Episodi 2 – Droonien hyökkäys

Tosielämän terminaattoreiden esikuvana voidaan pitää Arnold Schwarzeneggerin maanmiesten hyökkäystä Venetsiaan vuonna 1849. Itävaltalaisjoukot lähettivät piiritettyyn italialaiskaupunkiin 200 ilmapalloihin kiinnitettyä palopommia, joista tosin vain harva löysi perille. Osa räjähti ilmassa ja osa lensi hyökkääjien päälle. Aikalaiskertomukset kertovat venetsialaisten hurranneen ”ilotulitukseksi”. Miehitettömän sodankäynnin siemen oli kuitenkin kylvetty.

Kaapeliohjattuja Brennan-torpedoita käytettiin jo 1800-luvun lopulla, mutta eräs **Nikola Tesla** patentoi miehitettömän kulkuneuvon ohjausmekanismin vuonna 1898 ja esitteli radio-ohjatun torpedon. Risteilyohjauksen esi-isät puolestaan nähtiin 1918: esituantasteelle jääneet Hewitt-Sperry Automatic Airplane sekä Kettering Bug. Myös Ruston Proctor Aerial Targetia harkittiin käytettävän ilmalaivojen torjuntaan.

Vaikka varhaisten hankkeiden tulokset olivat vaatimattomia, viimeistään toisen maailmansodan aikana 1940-luvulla valmistui koko joukko eri tavoin kauko-ohjattuja lennokkeja ja ohjuksia. Yksi näistä oli saksalaisen *Fritz X* (1943), jota ohjattiin Kehl-Straßburg-radio-ohjauksella läheisestä lentokoneesta käsin.

Lennoykkien kehitys jatkui kylmän sodan ja Vietnamin sodan tiedustelukäytössä. Vuoden 1973 Jom kippur -sodassa Yhdysvallat

tuki Israelia toimittamalla sille muun muassa lennokkeja. Israel käytti drooneja hämäysmaaleina, jotta vihollinen haaskaisi kallisarvoisia ohjuksiaan. Jom kippur -sota johtikin yhdysvaltalaisen ja israelilaisten tutkijoiden yhteistyöhön lennokkekehityksessä.

Yksi tämän yhteistyön hedelmistä oli General Atomics MQ-1 -lennokki (1994), tuttavallisemmalta nimeltään Predator. Arnold Swarzenegger esiintyi samannimisessä elokuvassa seitsemän vuotta aikaisemmin, tosin elokuvan nimikkopahis ei ollut robotti eikä kyborgi, vaan avaruusolento. Hurjasta nimestä huolimatta tosielämän Predator oli kuitenkin alkuaan vain etäältä ohjattu miehittämätön tiedustelulennokki.

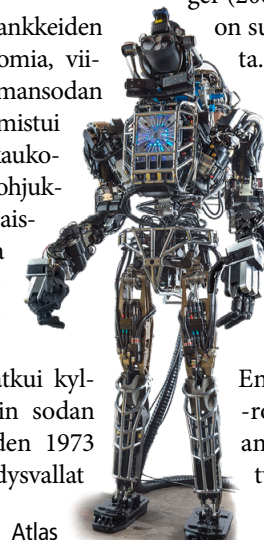
Erityisen vaaralliseksi Predatorin teki vuosituhannen vaihteen kehitystyö MQ-1A-versioksi, joka on aseistettu Hellfire-yläänohjuksilla. Aseistettu versio pääsi palveluskäyttöön syyskuun 11:nä 2001 terrori-iskujen jälkeen, ja siitä tuli terrorisminvastaisen sodan perusase. Taivaalla ampumisvalmiudessa pörräävä tappajarobotti, jonka ohjaajat istuvat nojatuoleissa kaukaisessa toimistossa, onkin se varsinainen tosielämän terminaattori.

Jonkinlainen ympyrä Venetsiasta nykypäivään sulkeutui vuonna 2016, kun Irakissa Mosulin taistelussa ISIS-terroristijärjestön kerrottiin käyttäneen kaupallisia nelikoptereita improvisoitujen räjähteiden kuljettamiseen. Kaikki sotilaslennot eivät kuitenkaan edelleenkään ole tappajarobotteja. Esimerkiksi Iranin kesäkuussa 2019 alas ampuma yhdysvaltalaislennokki oli aseistamaton Northrop Grumman RQ-4 Global Hawk (1998).

Predatorin on sittemmin korvannut pitkälti MQ-9 Reaper (ensilento 2001), MQ-1C Gray Eagle (2004) sekä Avenger (2009). Ohjusten lisäksi Avengeriin on suunniteltu HELLADS-laserasetta. Kehitteillä on myös lentotukialuskäyttöön sopiva taivassiiainen lennokki, Northrop Grumman X-47B.

Katalat keinokanuunat

Edellä mainitun RoboCop-elokuvan nimikkohahmo sai alkunsa, kun fiktiivinen Enforcement Droid Series 209 -robotti sai toimintahäiriön ja ampui robotin valmistaneen yrityksen työntekijän. Ratkaisuksi



Atlas

Tosielämän 343 Guilty Spark

343 Guilty Spark on lentävä robotti Bungien *Halo*-pelitrilogiassa – puhuva pallo, päähahmo Master Chiefin ajoittainen apuri ja vihulainen sekä kertakaikkisen ärsyttävä pikku pentele. Yhdysvaltain avaruusjärjestö NASA kehitti samantapaisia lentäviä pallorobotteja SPHERES-projektissaan vuosina 2000–2017, joskin alkuperäinen ulkonäköinspiraatio tuli *Tähtien sodan* valomiekkailudroonilta.

Aiemmin tänä vuonna SPHERESin seuraajaprojektin Astrobeen kuutiomaiset robotit lähetettiin kansainväliselle avaruusasemalle. Leijuvat Astrobeet muun muassa kuvaavat, inventoivat varastoja ja kuljettavat tavaraa. Kuin tribuuttina droonien historialle, avaruusmehiläisten nimet ovat: Bumble (kuvassa), Honey ja Queen.

Astrobee-lähdekoodi on avoimesti verkossa: github.com/nasa/astrobee



kehitettiin ihmiskyborgi hallitsemaan asekauden liikkeitä. Luoteja syytävät robotit ovat kuitenkin tosielämässä harvassa. Yleensä ne ovat osa jonkin sotakoneen, kuten laivan tai tankin, puoliautomaattista puolustusjärjestelmää, tai osa ohjuspuolustusjärjestelmää.

Ehkäpä kuuluisin näistä on yhdysvaltalainen Phalanx-lähipuolustusjärjestelmä, jota on käytetty laivoilla ja maatukikohdissa jo 1970-luvulta lähtien. R2-D2-robottia muistuttava tutkaohjattu järjestelmä hakee maalinsa automaattisesti ja syytää kuusiputkisella 20 millin Vulcan-kanuunalla 4 500 laukausta minuutissa kohteeseen. Vuonna 1996 japanilaisen hävittäjäaluksen JDS Yūgirin Phalanx teki ED-209:t ja ampui alas ystävällisen ”tunkeilijan”, yhdysvaltalaisen A-6 Intruder -lentokoneen.

Japanilaiset ovatkin kunnostautuneet robottiammuskeluissa. Kuluvan vuoden helmikuussa Japanin avaruusjärjestö JAXAn Hayabusa2-robottialus ampui 162173 Ryugu -asteroidia tannaalista valmistetulla luodilla – tällä kertaa tositarkoituksella. Asteroidi ei ollut ammuskelusta moksiskaan, mutta luodin irrottaman maa-aineksen toivotaan tuottavan tieteellisiä löytäjiä. Nebraskan Lincolnin yliopisto on kehittänyt samaan tarkoitukseen poralla varustettua NIMBUS-nelikopteria.

Lennokkien alasampumiseen tarkoitetut verkkosingot ovat jo vanha juttu, mutta miltä kuulostaisi singolla ampuva drooni? Venäläinen asevalmistaja UIMC esitteli vuonna 2016 singolla varustetun nelikopterin. Radiohärintäkään ei pysäytä tätä stalinin nokkahuilua, sillä kopterin kerrotaan olevan autonominen ilmaan päästyään. Villein venäläisrobotti on FEDOR (2017) – humanoidirobotti, joka ampuu pyssyllä, poraa poralla, ajaa autoa sekä tietenkin nostaa puntteja ja tekee punnerruksia.

Useimmat visiitit ampuvista roboteista jäävät kertaluontoisiksi julkisuustempuiksi, joista ei kuulla toiste. Ne ovat poikineet myös jekutuksia. Ihmismäisistä liikkeistään tunnetun robotin valmistajalla Boston Dynamicsilla todella oli Atlas-robotistaan ”karate-prototyyppi” Ian (2014), mutta kesällä viraalihitiksi noussut taistelurobotin kiusausvideo (vinkki: robotti saa tarpeekseen) oli pelkkää erikoistehostejekutusta. Venäläisen tanssivan Boris-robotin (2018) sisältä taas paljastui ihminen.



Tällä hetkellä näyttää siltä, että humanoidirobotit pysyvät laboratoriodien kuriositeettina siinä missä ne mechatkin. Sen sijaan lähitulevaisuudessa lennokkien ja automaattikanuunoiden kaveriksi sopii odottaa pieniä robottitankkeja, -laivoja ja -sukellusveneitä. Yksi yleinen robottilaiva on Hammerhead USV-T (2010) – kauko-ohjattu tykistömaali, kuinkas muuten. Telaketjurobotteja ovat esimerkiksi venäläinen Platforma-M ja polynimurifirman iRobot Warrior.

Tosielämän supersankarit

Arnold Swarzeneggerin terminaattori ei kuitenkaan elokuvan suomennetusta nimestä huolimatta ole synonyymi tuhoajalle. Swarzeneggerin terminaattorihahmosta kuoriutui myös lähes tuhoutumaton supersankari, heikkojen puolustaja.

Yksi tällainen arkipäivän supersankari on Shark Roboticsin Colossus-robotti (2017), joka kärrättiin paikalle, kun Pariisiin Notre Dame -kirkko kärähti viime keväänä. Ulkoisesti hieman viranomaisen käyttämiä pomminhävitysrobotteja muistuttava telaketjurobotti on peloton palomies: Colossus murtaa ovet, nuuhkii antureillaan ilmanlaadun ja sammuttaa tulipalot letkullaan.

Myös yhdysvaltain laivastolla on vastaava sammutusrobotti SAFFiR (*Shipboard Autonomous Firefighting Robot*). Mutta ehkä erikoisin viritys on japanilaisilla: Tohoku-yliopiston Dragon Firefighter (2018) on letkumallinen robotti, joka nostaa itsensä ilmaan vedenpaineen avulla ja ruiskujensa nostamana lentää kohteeseen sammuttamaan tulipalon ylhäältäpäin.

Robotit voisivat sopia erinomaisesti myös roskien keräämiseen ja vesistöjen jäteongelman ratkaisuun, mutta yksi haaste on niiden ohjastaminen. Altrubotsilla ajateltiin, että tämänhän voisi joukkoistaa: osoitteessa altrubots.com

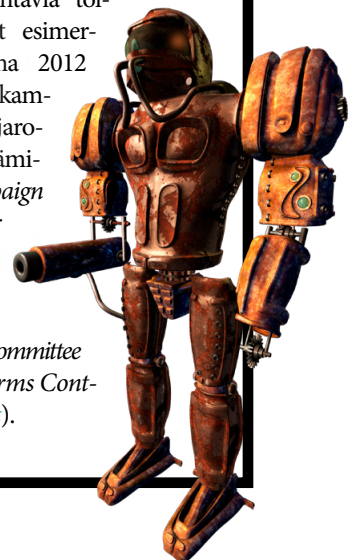
voit ohjata toisella puolella maailmaa ahkeroivaa Trashbot-roskienkeruurobottia oman kotisi rauhasta. 🤖

Linkit artikkelin robotteihin Skrollin verkkojatkoilla: skrolli.fi/numerot

Yhteistä useimmille edellä mainituille tosielämän terminaattoreille on ihmiskontrolli – niitä ohjataan etäältä käsin. Seuraava polvi autonomisia ihmeaseita on kuitenkin työn alla. Ne tunnetaan englanninkielisellä lyhenteellä LAW (*Lethal Autonomous Weapons*).

Yhdistyneet kansakunnat onkin havahtunut säätelykysymykseen. YK:n Eräitä tavanomaisia aseita koskeva yleissopimus on perinteisesti pyrkinyt hillitsemään sellaisten erityisen ongelmallisten aseiden käyttöä kuten miinat ja sokaisevat laserit, mutta viime vuosina työn alle on otettu myös kysymys autonomisista asejärjestelmistä.

Myös valtiohallinnon ulkopuoliset järjestöt ovat aktivoituneet. Merkille pantavia toimijoita ovat esimerkiksi vuonna 2012 perustettu kampanja tappajarobottien kieltämiseksi (*Campaign to Stop Killer Robots*, stop-killerrobots.org) sekä *International Committee for Robot Arms Control* (icrac.net).



Markku Alasen MIKROBITTI

Skrollin henkisen esikuvan, MikroBitti-lehden, entinen päätoimittaja Markku Alanen kuoli toukokuun lopussa yllättäen. Hän oli se supermario-viiksien takaa pääkirjoituksessa hymyilevä mies, Bitin alkuvuosien nimimerkki Mervyn J. Snafu ja minun isäni.

Ninnu Koskenalho

Olin viisivuotias, kun isä meni töihin minua kolme vuotta nuorempaan MikroBittiin. Siihen asti isän työ oli tarkoittanut Lundia-hyllöjen sivuista roikkuvia mustavalkoisia pitkiä valokuvaliuskoja, joissa esiintyi jännittäviä, kromosomeiksi kutsuttuja muotoja. Isä väitteli molekyyli-genetiikan tohtoriksi Helsingin yliopistosta vuonna 1985, mutta lupaava tutkijan ura vaihtui yllättäen lapsen näkökulmasta hauskeempaan hommaan.

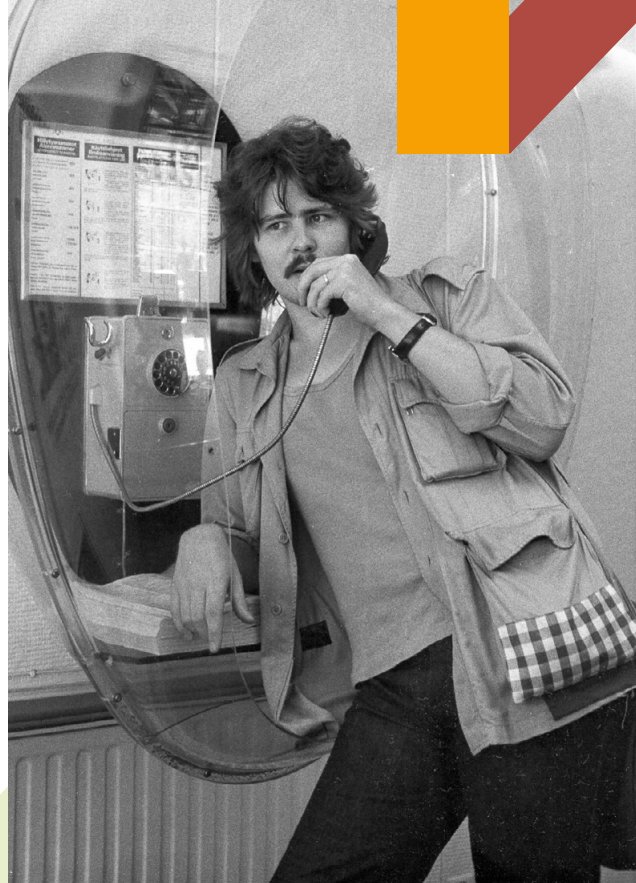
Vuonna 1984 perustettu MikroBitti oli pieni harrastajalehti, joka oli saanut alkunsa vuotta aikaisemmin toisen lehden liitteenä. Uudessa lehdessä idearikas ja visionäärinen isäni toimi ensimmäiset pari vuotta toimittajana. Vahvojen näkemysten ja niiden suunnitelmallisen toteuttamisen vuoksi hän siirtyi ensin toimituspäällikön ja vuonna 1994 päätoimittajan paikalle.

Markun johdolla Bitin levikki kasvoi suomalaiselle erikoislehdelle käsittämättömiin mittasuhteisiin, yli sataantuhanteen. Mukava seuraus isän työstä oli koko lapsuuteni ja teini-ikäni lävistänyt tietokoneiden, pelikonsolien, pelien ja muiden kiinnostavien vimpaimien ylenpalttinen läsnäolo kotonamme.

Jos 80-luvun Bitti oli harrastajien paperinen tee-se-itse-hacklab ja kokonaisen tietotekniikkaosaajien sukupolven yhteinen toiminnallinen päiväkotio, 90-luvun Bitti oli nettiajan menestyksekkäs edelläkävijä. Lehden lopettaminen oli 90-luvun alussa lähellä, kun samoista lukijoista kilpaili yhtäkkiä julkaisu jos toinenkin. Aina pitkälle tulevaisuuteen tiiraava isäni tiesi, että suurelle yleisölle vielä tuntematon Internet oli ihmiskunnan väistämätön seuraava askel. Itselleen tyypilliseen tapaan hän katsoi olemassa olevia verkkopalveluita ja tuumi, että lienee fiksuinta tehdä itse parempi.

Vaikka Internet ei 90-luvulla ollut enää oikeasti uusi keksintö, se oli edelleen nörttien hommaa. Yhteiskunnan digiloikka ei ollut kunnolla alkanut. Markku tiesi, ettei lehden nettiulottuvuus voinut tarkoittaa pelkästään paperilla julkaistavien sisältöjen siirtoa verkkoon sellaisenaan. Hybridimedia tarkoitti sitä, että verkko-Bitti ei ollut ainoastaan lehti digitaalisessa muodossa. Se oli tuore konsepti, joka pyrki aidosti hyödyntämään netin tuomia mahdollisuuksia sosiaalisuuteen ja vuorovaikutteisuuteen.

”Samat asiat motivoivat ihmistä netissä kuin muuallakin elämässä”, sanoi isä. Hänen filosofiansa verkkoulottuvuuden luomisessa pohjasi



1985 tuleva päätoimittaja keskittyi väitöskirjaansa, Bitti täytti vuoden ja kirjoittaja sai isältä 4-vuotislahjaksi Donkey Kong II -elektronikkapelin.

käyttäjien tarpeiden, motivaatioiden, nautintojen ja palkitsevyyden kokemusten sekä itsekokeiden taipumusten ymmärtämiseen. Aikana, jolloin palvelumuotoilua verkko-ympäristöille ei ollut vielä ehditty edes ajatella, Markku suunnitteli oikeisiin tarpeisiin vastaavia, käyttäjälleen mielihyvää tuottavia palveluita. Suurin osa hänen idearikkaan mielensä tuotoksista ei koskaan nähnyt päivänvaloa, mutta monet niistä tulivat vuosia myöhemmin vastaan muiden toteuttamina.

MikroBitti oli niin historiallinen projekti pitkälti siksi, että se oli oikeassa paikassa oikeaan aikaan. Se oli kuitenkin myös paljon enemmän kuin onnekas sattuma. Toimituksessa, jossa itsekkin työskentelin lyhyesti 2000-luvun alussa ja vierailin lapsena useasti (Bitin legendaarisessa labrassa Monkey Islandia pelaten, sivusilmällä tietokoneita purkavia ja rakentelevia ihmisiä tarkkaillen), oli innostava yhteishenki ja tekemisen meininki. Yhteisöllisyys ja tee-se-itse-henki olivat edelleen voimissaan uuden vuosituhannen koittaessa, vaikka fokus olikin siirtynyt yhä voimakkaammin kohti nettiulottuvuutta.

Markun johdolla Bitistä tuli lehtialan menestystarina, ja hybridimedia-konsepti rikkoi uusia ennätyksiä. Eräälle sukupolvelle niin muistiinpainuva BBS-kulttuuri jäi nettiaikaan siirtyessä suurelle yleisölle täysin tuntemattomaksi ilmiöksi. Sen loppurajähdyksenä syntyi vuonna 1994 isäni innovoima maailman suurin purkki MBnet, joka suljettiin vasta 2002. Hybridimedian myötä toimitus kasvoi kattamaan lehteä tekevän vanhan koulukunnan rinnalle uusmediaosaston, joka oli vastuussa lehden sähköisestä ulottuvuudesta. Samoin kuin lehden lukijoista tuli sen tekijöitä, MBnetin käyttäjistä tuli sen ylläpitäjiä ja hybridi-Bitin luoja.

Vuonna 2004 MikroBitti juhlisti komeaa 20-vuotista taivaltaan. Sen kunniaksi julkaisija nakkasi isäni pihalle päätoimittajan pallilta rumalla ja epäkunnioittavalla tavalla, joka shokeerasi silloista toimitusta, eikä varmasti näin jäl-

kikäteenkään tee ketään ylpeäksi. Seurauksena MikroBitin huimaava menestystarina kääntyi nopeaan ja draaattiseen laskuun.

Näppäimistö tulessa

Seuraavalla aukeamalla oleva, vuonna 2013 kirjoitettu kolumni on isäni sana siihen, mitä Bitille todella tapahtui. Koska Skrolli oli vielä uusi ja paikkaansa hakeva tulokas lehtialalla, Markkua pyydettiin muokkaamaan tekstistään vähemmän hyökkäävä. Kiltimpi versio julkaistiin Skrollin numerossa 2013.2, ja sen voi pdf-muodossa lukea osoitteessa skrolli.fi/numerot.

Kolumnin haikean kettuilevaan sävyyn on hyvät syyt. Isääni ei kismittänyt työstä luopuminen, sillä hän oli kyllästynyt julkaisijan kanssa asioimiseen. Katkeruus suuntautuu siihen, että – ja kuinka – Bitti tuhottiin.

Markun aikana Bitti oli sisältövetoinen projekti. Se halusi palvella lukijoiden sellaisiakin tarpeita, joita nämä eivät osanneet sanallistaa. Se halusi olla lukijan kaveri, tehdä ja oppia yhdessä. Levikin kasvattaminen oli selkeä tavoite, joka saavutettiin kaikki rajaidat paukkuun. Levikkiä ei kuitenkaan tavoiteltu osakkeenomistajien lompakot mielessä, vaan aidosti tyytyväisten lukijoiden ja käyttäjien vuoksi.

Isä koki, että 2000-luvun puoliväliin mennessä tälle tuli loppu. Melkein 10 vuotta uransa päättymisen jälkeen hän kirjoitti Tyhjästä Joukosta, ”kasvavasta laumasta konsultteja, asiantuntijoita, pikkupomoja, päälliköitä ja johtajia kollektiivisessa suojatyöpaikassaan hokemassa mantraa keisarin kauniista vaatekerrasta”, syyttäen näitä kadettömyydestä, tyhmyydestä ja vastuuntunnon puutteesta. 2000-luvun alkupuolella lehden julkaisijan johdossa tapahtui muutoksia, joiden seurauksena ”journalismi jäi rahanhimon jalkoihin ja [Aatos] Erkon perintö tuhottiin”, kuten isäni synkkänä huokaili. ”Viimeisetkin oikeat journalistit savustetaan täältä pian ulos”, hän murehti.

Tämä oli aikaa, jolloin printtimedia kappaili digitaalisen monsterin kanssa. Nouseva voima uhkasi syödä medialta sekä tulot että jatkuvuuden. Kun tiedon valtatieta rakennettiin kovaa vauhtia osaksi kansalaisten jokapäiväistä elämää, joutuivat perinteiset mediat keksimään itselleen vauhdilla uusia menopelejä kyydissä pysyäkseen. Huolimatta siitä, että MikroBitti oli toiminut tässä prosessissa tiennäyttäjänä, isäni elämäntyö huuhdeltiin nopeasti viemäriin. 2000-luvun alun Bitti ehti vielä kannatella kokonaisen

sukupolven nettiikaan. Kun lehden visionäärinen kapteeni jätettiin rantaan, laiva ajoi karille.

Meidän isä

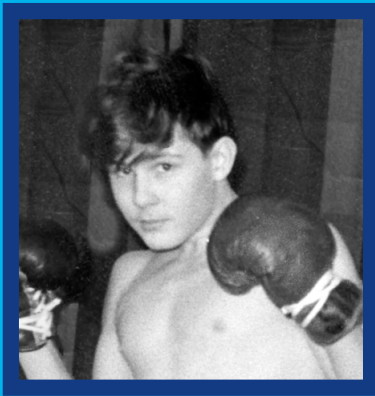
Dollarinkuvat silmissä viipottaminen kyrpi Markkua uskomattoman paljon. Ahneus ja häikäilemätön oman edun tavoittelu saivat hänet takuuvarmasti raivoihinsa. Hän viittaa kolumnissaan useampaan kertaan siihen, että Mammonin alttarilla onanointi oli se, mikä vei projektilta hengen. Isäni piti yleisestikin kaupankäyntiä huijauksena ja halusi paljastaa, miten paljon rahaa jää erilaisille, hänen mukaansa turhille, välikäsille. Tästä mielenmaisemasta syntyi muun muassa MBnetin Hintaseuranta-palvelu. Rahanahneus näyttäytyi hänelle pahantahotoisena ja moraalittomana, maailman tuhoavana asenteena.

Voittojen sijaan Bitin johtotähti oli näkemys siitä, että itse tekeminen on hauskaa. Sen sijaan, että kaikki hankittaisiin valmiina, keskityttiin rakentamaan, korjaamaan ja tuunaamaan – kuten meillä kotona. Isä halusi saada lukijat innostumaan asioista ja tuntemaan sekä lehden että ideat omikseen. Häntä kalvoi ihmisten potentiaalain käyttämättä jättäminen, ja hän oli tunnettu epätavallisesta kyvystään saada ihmiset uskomaan omiin kykyihinsä ja ideoihinsa.

Minulle Bitti oli sisarpuoli: isän huolella vaalittu, paljon huomiota saava lapsi. Kun isä toi 90-luvulla työn alla olevan numeron paperisia vedoksia kotiin korjattavaksi, hänellä oli tapana käydä kansivedokset läpi kanssani ja pyytää minua kertomaan, mitä hyvää ja huonoa niissä näin. Aiheesta käymämme keskustelut tuskin vaikuttivat kansien lopulliseen ulkonäköön, mutta minun käsitykseeni näkemysteni uskottavuudesta kyllä.

Skrollin alun perin julkaisema kolumni jätti pois myös oheisen version Nokia-ruodinnan. Poistettujen osien tilalle isä kirjoitti kehaisun Skrollin kaverilliselle, yhteisölliselle tee-se-itse-meiningille. Hän arvosti Skrollin henkeä, jossa on paljon 80-luvun ja 90-luvun alkupuolen Bittiä, sekä lehden epäkaupallisuutta ja eräänlaista itsepintaista hulluutta. Isän sydämessä oli aina soppa varattuna intohimosta syntyneille, vastavirtaan uiville, tinkimättömille projekteille, joiden motiivi on muualla kuin pankkitilin paisuttelussa.

Markku Alasen MikroBitti kasvatti Suomeen mikroilijoiden sukupolven, ja sitä kautta minun erityinen, viisas, itsepäinen ja lempeä isäni oli vähän teidänkin isänne. 🏠



Markku Alanen oli MikroBitti-lehden päätoimittaja vuosina 1994–2004. Markku kirjoitti Skrollille tämän kolumnin vuonna 2013 ja siitä julkaistiin lyhennetty versio Skrollissa 2013.2. Julkaisemme nyt Markun perheen luvalla koko kolumnin, sellaisena kuin Markku sen alunperin tarkoitti. Teemat ovat kuusi vuotta vanhoja, mutta ajatukset ikuisia.

Kiitos, Markku.

MikroBITTI in memoriam

...ja vähän Nokiakin

Skrollin lukijoista useimmilla lienee jonkinlaisia omakohtaisia kokemuksia MikroBITTI-lehdestä. Aktiivisimmat olivat mukana Bitin tekemisessä jo vuoden 1984 ensimmäisestä numerosta alkaen – siis lukijat lehden teossa. Nykyisin tällaista kutsuttaisiin joukkoistamiseksi.

Bitille lukijoiden osallistuminen ei ollut ainoastaan lehden sisällön määrittämistä vaan myös sisällön tuottamista. Toimitus ja avustajat olivat yötyöllistettyjä kasibittisiä ohjelmallistauksia kaseteilta purkaessaan. Bitin mahtavan alkumenestyksen taustalla oli ensisijassa juuri uudenlainen lukijasuhde. Jako tyhmiin lukijoihin ja teräviin toimittajiin kuopattiin kuuden jalan syvyyteen. Oltiin kaikki enempivähempi samaan hiileen puhaltavaa porukkaa.

Bitti oli asemoitu harrastelehdeksi. Erikoista kuviossa oli se, että harrastuksesta tuli useille ammatti. Ajan mittaan yhä suurempi osa lukijoista oli joko valmiiksi tietotekniikkahommissa tai kovaa vauhtia matkalla moiisiin. Moni lukija sai lehdestä enemmän potkua tietotekniikan saloihin syventymiseen kuin konsanaan muusta atkopastuksesta. Vahva yhteisöllisyys piti samassa veneessä niin aloittelijat kuin edistyneemmät bitin nyplääjät. Bittiin mahtuivat niin koodarit, pelaajat kuin rakentelijatkin, vaikka välillä aiheuttivatkin ahdistusta toisilleen.

Kova kilpailu lukijoista oli päivän sana tultaessa 1990-luvulle. Bitin asema oli tukala ja lehden toiminnan lopettaminen oli hilkulla. Kun alkuvuosina vain Printti-lehti kalasteli menestyksekkäästi samoilla vesillä Bitin kanssa, niin nyt täytyi tilaajia jakaa niin Tiekkarille kuin mikroPC:llekin. Osa Bitin lukijoista alkoi livetä ruodusta ja käydä vieraisa. Ammatillisten ambitioiden täytyessä yhä useampi koki arvostuskysymyksenä "harrastelehden" nimeen

vannomisen, kun tarjolla oli ammatilehtiäkin.

Eikä tässä vielä kaikki. Ulkoisten kilpailijoiden lisäksi Bitti järjesteli kotoperäisesti markkinoille C=lehden, Pelit-lehden ja vielä kotiPC:n. Kaikki nämä tavoittelivat osittain samoja lukijoita ja rokottivat tilaustuloja, joilla Bitin kulut kuitattiin. Tilanne oli se, että jotain tarttis tehdä ja vanhat konstit oli kaikki käytetty. Minulle esitettiin kysymys: "Pannaanko pillit pussiin Bitin osalta vai voitko oikaista jotenkin lehden kuolonkierteen?" Lupasin hoitaa homman kotiin. Asian järjestämiseksi pähkäsin Bitille hybridimedia-konseptin. Ego, ego. : [])

Internet oli vielä harvojen ja valittujen temmellyskenttä 90-luvun alussa. Web oli vasta tuloillaan ensimmäisen Mosaic-selaimen myötä. Ei ollut Irc-galleriaa tahi Facebookia. Verkkoyhteisöt roikkuivat vielä puhelinlangoilla BBS-palvelimissa. Suunta oli kuitenkin selvä: Netti tulee ja Netti tappaa. Parempi hypätä rattaille mahdollisimman alussa. Pantiin pystyyn MBnet – Bitin sähköinen ulottuvuus – ja kutsuttiin kaikki lukijat mukaan maailman suurimpaan purkkiin, eräänlaiseen paikalliseen internetiin. Nettiajan voidaan syystä katsoa alkaneen Suomessa 1994 hetkellä, jolloin puhelinverkot kaatuivat MBnettiin pyrkivien painosta.

MBnet ja BITTI muodostivat yhdesä uudenlaisen hybridimedian. Netti ja paperilehti pyrittiin nivomaan yhteen siten, että molemmat palvelisivat parhaalla tavalla lukijaa. Nettipuoli ei ollut mikään pakollinen kasvannainen vaan tasavertainen paperilehden kanssa. Lehdessä käsiteltävien aihealueiden lukumäärä ei netin myötä suinkaan vähentynyt vaan kasvoi rajusti. Oli pakko lisätä sivumääriä ja suorittaa karsintaa vanhojen, vakiintuneidenkin sisältöjen suhteen. Lukijoiden hyväksyntä muutoksille ostettiin tarjoamalla vanhan tilalle riittävän paljon mielenkiintoista uutta sisältöä.

Uusi MikroBITTI osoittautui kaikilla mittareilla menestykseksi. Lukijat olivat tyytyväisiä ja saivat entistä helpommin äänensä kuuluviin netin välityksellä. Toimitus ja avustajat olivat ylpeitä aikaansaannoksistaan. Levikit kasvoivat hurjaa vauhtia ja kilpailijat karisivat kyydistä. Bitistä kasvoi väkilukuun suhteutettuna maailman laajalevikkisin alan lehti. Toimittajat ja lehden avustajat edustivat alan terävintä kärkeä niin ammattitaidoltaan kuin lehden sisällön kehittämiseen sitoutumisessaan. MikroBITTI oli yhteinen asia ja sen tekeminen oli hauskaa ja antoisaa. Palkitsevaa se oli myös lehden omistajalle. Lehden kate nousi 40 % tuntumaan (esimerkiksi 10 miljoonalla pyörítettävä bisnes antaa voittoa 4 miljoonaa vuodessa). Lehden talous ei ollut pätäkääkään kiinni mainosmyynnistä – tilanne, josta useimmat lehdet voivat vain unelmoida. Sähköisellä puolella MBnet oli arvostetuimpien nettisaittien top kymppissä.

Lopulta kävi niin kuin sadussa, joka kertoo kultamunia munivasta hanhesesta. Ahneus iski kustantajaan, joka julisti itsensä lehden sisältövastaavaksi ilman alkeellisintakaan ymmärrystä sen enempää lehdestä kuin netistäkään. Bitin tekijöille oli aina ollut selvää, että lehden lukijoille annettua lupausta ei saa pettää – lupausta siitä, millainen lehti on. Tämä lukijan kunnioitus sai lentää romukoppaan, kun Tyhjä Joukko* alkoi ohjata lehteä. MikroBitti luopui keskeisimmästä kivijalastaan – lukijoiden kunnioittamisesta – ja lopputulos on kaikkien henkilökohtaisesti arvioitavissa. MikroBITTI on kuollut ja sen tilalle lanseeratun mB-lehden tulevaisuus on pitkälti sen varassa, että lehti löytää riittävästi itsensä kaltaisia sisältöiltään tyhjiä, uusia tilaajia. Sellaisia, jotka eivät heti huomaa tulevansa kusetetuiksi.

Muutos on luonnollista meni se suuntaan tai toiseen. Jos BITIN kohdalla onnistuttiin kämmäämään oikein ruhtinaallisesti, niin ei paremmin mene Nokiallakaan. **Siilasmaan Risto** hoiti aikanaan Bitin ohjelmallistauksia tunnollisesti ja hyvin. Nokian hallituksen puheenjohtajana näytöt on vielä antamatta – ja taitavat sille tolalle jäädäkin. Nokian vene on ajettu korskeasti kiville. Kun Nokia otti aikanaan Applelta ja Samsungilta kunnolla turpiinsa, pelastajaksi viritettiin Windows-puhelinta pitkän

odotuksen verran. Nokia sai aikaiseksi Lumia 920 -kapulan.

Niin minä kuin miljoonat muutkin odottivat jotain wow-mullistavaa, jota muilla valmistajilla ei ole tarjota. Ilmeni pian, että kyseessä oli pelkästään yhtä hyvä puhelin kuin kilpailijoillakin. Tästä häkelymättä oletin Nokian noudattavan harkittua strategiaa ja putsaaavan pöydän hinnoittelupolitiikalla. Varmistettaisiin valmistus ja myytäisiin Lumiaa sellaiseen hintaan, että kaikki paitsi paatuneimmat iPhone-hörhöt kiirehtivät kauppaan. Markkina-asemien varmistuttua olisi kasa keinoja tuottaa taas tyydytystä osakkeenomistajille. Näinhän ei sitten menetelty. Kyseessä oli ainutkertainen mahdollisuus, joka tuskin tulee toistumaan. Iskikö Tyhjä Joukko taas?!

Ja lopuksi käsi sydämelle lukija. Ostaisitko tuntemattomalta tai huonomaineiselta älypuhelinvalmistajalta luurin, joka maksaa yhtä paljon kuin hyvämaineisten valmistajien lippulaivamallit? Vaatii varmaan fannittamista. Jos vastasit myöntävästi, saatat sopia Nokialle joukon jatkoksi. ;) Siellä ihmetys on suuri, kun Lumia ei myy?! 🐼

Markku Alanen, FT
kirjoittaja ohjasi MikroBittiä ennen Tyhjä Joukkoa

** Tyhjä Joukko = kasvava lauma konsultteja, asiantuntijoita, pikkupomoja, päälliköitä ja johtajia kollektiivisessa suojatyöpaikassaan hokemassa mantraa keisarin kauniista vaatekerrasta. Koska joukko on tyhjä ja sillä on suhteettomasti valtaa, se saa paljon tuhoa aikaan – ei ilkeyttään, vaan silkkaa kykenemättömyyttään.*

*"...ei ollut Irc-galleriaa
tahi Facebookia.
Verkkoyhteisöt roik-
kuivat vielä puhelin-
langoilla BBS-palveli-
missa. Suunta oli
kuitenkin selvä:
Netti tulee ja
Netti tappaa..."*



Säteenaskellus ja etäisyyskentät Grafiikkamoottori pienoiskoossa

SDF-primitiivejä
Inigo Quilezin
Shadertoy-
demonstraatiossa
([shadertoy.com/
view/Xds3zN](http://shadertoy.com/view/Xds3zN))

Teksti ja kuvat: Jarno Niklas Alanko

*Haluaisitko tehdä oman neljän kilotavun intron, mutta et tiedä mistä aloittaa?
Tutustutaan pienten introjen erikoistekniikkaan – säteenaskellus etäisyyskentillä.*

Kokorajoitettujen grafiikkademot eli introt ovat hankala taiteenlaji. Näissä ohjelman koko on rajoitettu esimerkiksi yhteen tai neljään kilotavuun, jolloin tavanomaisen grafiikkamoottorin käyttö on mahdollista.

Kolmiulotteiset mallit esitetään tietokoneissa yleensä kokoelmana yhteen linkitettyjä kolmiota (*triangle mesh*). Avaruuden kolmio on teoriassa helppo valaista ja projisoida näytölle, mutta kolmiomallin määrittely ja piirto vie kuitenkin yleensä liikaa kooditilaa pienimmissä introissa. Pienet kolmiulotteiset introt käyttävät sen sijaan tyypillisesti jonkinlaista säteensuuntausta (*raycasting*).

Säteensuuntauksessa asetetaan kolmiulotteiseen avaruuteen kuvitteellinen kamera ja ikkuna, ammutaan kamerasta valonsäteitä ikkunan jokaisen ”pikselin” läpi ja katsotaan mihin kappaleeseen se osuu, jos osuu. Valaistusta ja varjoja voidaan laskea määrittelemällä avaruuteen valonlähteitä ja seuraamalla säteitä valonlähteistä kappaleiden pinnoille.

Valonsäteen ja yksinkertaisten kappaleiden, kuten pallojen ja kolmioiden, leikkauspisteet ovat melko yksinker-

taisia laskea¹, mutta pelkillä palloilla ei pääse vielä pitkälle, eivätkä kolmioko-koelmat edelleenkään tule kysymykseen pienimmissä introissa. Tarvitaan tiiviimpi tapa määrittellä maailma.

Tässä jutussa tutustutaan eräeseen pienten introjen perusmoottoriin. Kyseessä on kahden tekniikan yhdistelmä: etumerkilliset etäisyyskentät (*signed distance fields, sdf*) ja säteenaskellus (*raymarching*). Etäisyyskenttä määrittelee maailman ja säteenaskellus piirtää sen. Kutsumme tätä yhdistelmää RMSDF-moottoriksi (*raymarching signed distance fields*).

Moottorin perusidea

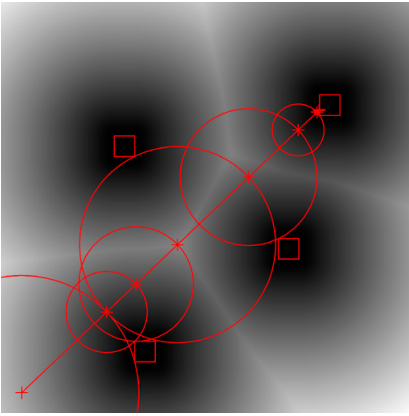
Aloitetaan etäisyyskentistä. Etumerkillinen etäisyyskenttä on funktio, joka ottaa avaruuden pisteen ja palauttaa etäisyyden lähimpään pintaan. Jos etäisyys on positiivinen, ollaan kappaleen ulkopuolella, ja jos negatiivinen, ollaan kappaleen sisällä. Esimerkiksi funktio $f(x,y,z) = \sqrt{(x^2 + y^2 + z^2)} - 3$ määrittelee maailman, jonka origossa on pallo, jonka säde on kolme. Nyt esimerkiksi

$f(1,0,0) = -2$ tarkoittaa sitä, että piste (1,0,0) on pallon sisällä ja etäisyydellä 2 lähimmästä pinnasta. Pisteet, joissa funktio palauttaa nollan, ovat täsmälleen pallon pinnalla. Introissa etäisyyskenttä ottaa yleensä myös parametrina kulloisenkin ajanhetken, jotta kenttää voidaan muokata ajan kuluessa.

Etäisyyskentän kuvaama maailma voidaan piirtää säteenaskelluksella. Tämä on normaalista säteensuuntauksesta numeerinen versio, jossa säästyään säteiden ja kappaleiden leikkauspisteiden täsmälliseltä ratkaisemiselta. Säteenaskelluksessa marsitaan pitkin valonsäteitä sopivan kokoisin askelin, kunnes pinta tulee vastaan. Askelten pituudet saadaan etäisyyskentästä: jos etäisyyskentän mukaan lähin pinta on x yksikön päässä, niin on varmasti turvallista kulkea matka x pitkin valonsädettä ilman, että törmätään mihinkään. Tällaisia askelia toistetaan vaikka 50 kertaa. Jos etäisyyskentän antama arvo lopussa on tarpeeksi pieni, vaikka alle 0,01, olemme osuneet kappaleen pintaan. Muussa tapauksessa oletamme, että säde ei osunut mihinkään (kuva 1).

Kun säde osuu kappaleeseen, lasjetaan osumakohdan kirkkaus ja väri jollain valaistuskaavalla. Nämä kaavat

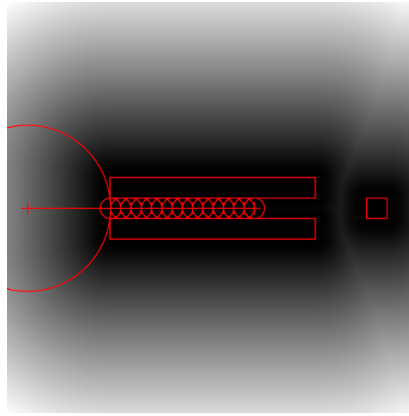
¹ Perinteistä säteensuuntausta ja -seurainta on käsitelty Skrollissa 2017.4 artikkelissa *Tee-se-itse raytracer*.



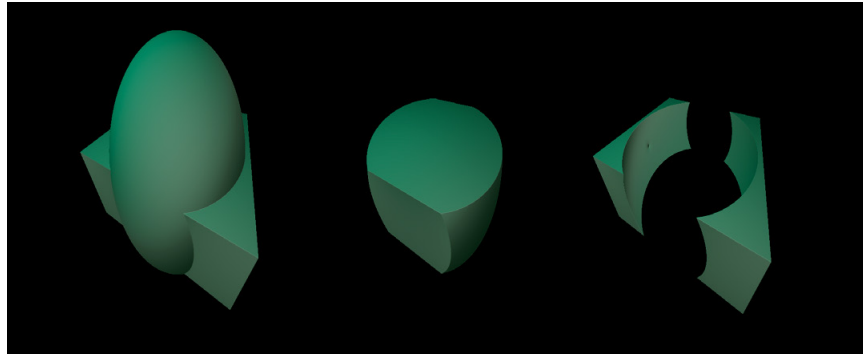
Kuva 1. (yllä) RMSDF-algoritmi kahdessa ulottuvuudessa. Maailmassa on neljä neliökappaletta. Pohjalla on etäisyyskenttä harmaalla: mitä kirkkaampi pikseli, sitä suurempi etäisyys. Valonsäde lähtee vasemmasta alanurkasta kohti yläoikeaa. Jokaisella askeleella mitataan aluksi etäisyys d lähimpään kappaleeseen. Tämän jälkeen tiedetään, että jos tämänhetkisen pisteen ympärille piirretään d -säteinen ympyrä, sen sisälle jää pelkkää tyhjää, joten d yksikön mittaiset askeleet ovat varmasti turvallisia. Algoritmi ottaa d yksikön pituisen askeleen sädettä pitkin, ja sama toistetaan uudestaan. Kuvassa valonsäde pääsee jo muutaman askeleen jälkeen liki lopulliseen päämääräänsä. Kolmessa ulottuvuudessa tätä algoritmia kutsutaan myös nimellä palloaskellus (sphere tracing).

vaativat yleensä pinnan normaalin, eli vektorin, joka on kohtisuorassa pintaa kohden. Eräs etäisyyskenttien hienous on se, että pintojen normaalit saadaan laskettua ”ilmaiseksi”: pinnan normaali on se suunta, jossa etäisyyskenttä kasvaa voimakkaimmin, eli etäisyyskentän *gradientti*. Gradientti on derivaatan yleistys usean muuttujan funktioille. Sen täsmällinen laskeminen voi olla työlästä, mutta siitä saadaan hyvä numeerinen arvio ottamalla minimaalisen pieni askel avaruudessa ja katsoamalla, kuinka etäisyyskenttä muuttuu. Tässä on olennaista, että etäisyyskenttä määritellään negatiiviseksi kappaleen sisällä.

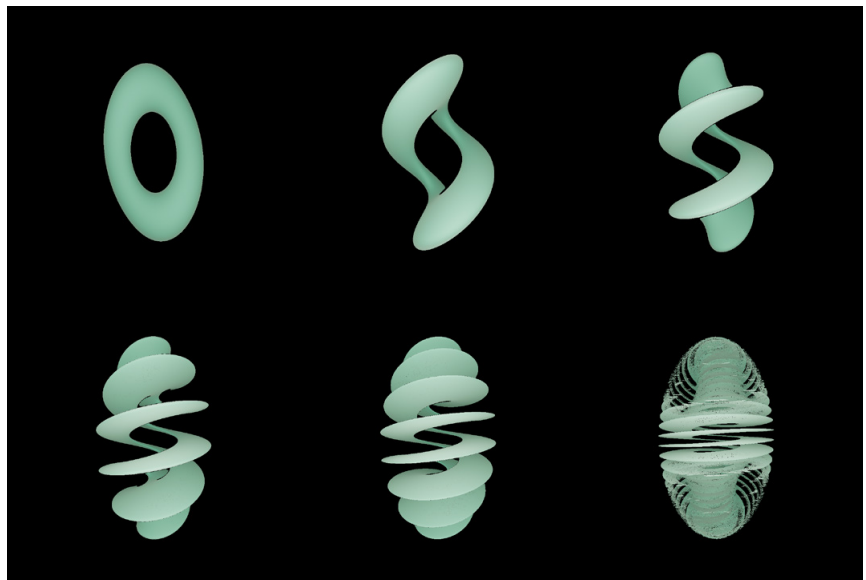
Etäisyyskentän määrittelyssä ei välttämättä tarvitse olla matemaattinen velho, vaan riittää yhdistellä valmiita rakennuspalikoita eli ”primitiivejä”. Internetistä löytyy valmiita kaavoja erilaisille primitiiveille, ja paras resurssi on ehdottomasti SDF-guru **Inigo Quilez**in verkkosivut (iquilezles.org/www/articles/distfunctions/dist-functions.htm). Sivuilta löytyvät kaavat esimerkiksi palloon, suorakaiteeseen, donitsiin, sylinteriin ja kartioon. Primitiivejä voi liikuttaa, pyörittää ja yhdistellä yksinkertaisilla kaavoilla.



Kuva 2. Pitkät kapeat tunnelit ovat haastavia RMSDF-moottorille. Niissä ei päästä ottamaan pitkiä askelia, koska seinät ovat aina lähellä valonsädettä. Kuvassa on tällainen ongelmatapaus. Säde on lähtenyt vasemmalta ja ottanut jo 15 askelta, eikä ole vielä osunut yhteenkään pintaan, joten moottori voisi jo tulkita säteen menneen ohi kaikesta. Tässä ei voida tietää, olisiko jossain kaukana edessä ollut kuitenkin kappale, johon säde olisi lopulta osunut. Reaaliaikainen moottori ei voi kuitenkaan askeltaa kovin monta kertaa, koska muuten ruudun päivitysnopeus kärsii.



Kuva 3. SDF-primitiiveille on mahdollista tehdä erilaisia joukko-operaatioita yksinkertaisilla kaavoilla. Kuvissa on yhdistelty pallon ja suorakulmisen särmiön primitiivejä käyttäen listauksen 2 operaatioita. Vasemmalla näkyy unioni, keskellä leikkaus ja oikealla on suorakulmaisesta särmiöstä on vähennetty pallo. Oikeanpuoleisimmassa kappaleessa näkyvä pieni ”lommo” on jonkinlainen bugi tai säteenaskelluksen artefakti.



Kuva 4. Kuvasarja demonstroi etäisyyskenttää vääristävää kierre-efektiä donitsille. Mitä suurempi kierre on, sitä enemmän kenttä vääristyy, ja lopussa säteet alkavat mennä jo läpi pinnoista.

Sivustolla kaavat on annettu suoraan koodina, jonka voi helposti kopioida omaan koodiin.

Seuraava taso: vääristävät muunnokset

Etäisyyskenttää voidaan venyttää ja vääristää mielenkiintoisin lopputuloksin.

Kappaleiden pintoja, eli etäisyyskentän nollakohtia, voidaan muovata laittamalla etäisyyskenttää jonkinlaisen muunnoksen läpi (kuva 4). Tässä saattaa tosin käydä niin, että etäisyyskentän antamat arvot eivät enää pidä paikkaansa pintojen ulkopuolella. Etäisyyden aliarviointi onneksi vain hidastaa säteenaskelluksen

```

// Origokeskeinen pallo.
// p = piste, josta etäisyys lasketaan.
// s = pallon säde.
float sdSphere(vec3 p, float s) {
    return length(p)-s;
}

// Origokeskeinen suorakulmainen särmiö eli laatikko.
// p = piste, josta etäisyys lasketaan.
// s = laatikon leveys, korkeus ja syvyys.
float sdBox(vec3 p, vec3 b) {
    vec3 d = abs(p) - b/2.0;
    return length(max(d,0.0)) + min(max(d.x,max(d.y,d.z)),0.0);
}

```

Listaus 1. Esimerkkejä SDF-primitiiveistä: origokeskeinen pallo ja laatikko. Pallon kaavassa etäisyys on yksinkertaisesti syötepisteen etäisyys origosta miinus pallon säde. Laatikon määrittelevässä kaavassa $\text{abs}(p)$ peilaa ensin syötepisteen kolmiulotteisen koordinaatiston positiiviseen osaan ja $b/2$ vähentäminen siirtää koordinaatistoa niin, että laatikon positiivinen yläkulma on origossa. Paluuarvon ensimmäinen termi laskee etäisyyden laatikon ulkopuolella ja toinen termi negatiivisen etäisyyden pallon sisäpuolella.

```

// Siirto offset-parametrin määräämään suuntaan
vec3 translate(vec3 p, vec3 offset, sdf3d primitive){
    return primitive(p - offset);
}

// Pyörittäminen annetun akselin ympäri annetun kulman verran
vec3 rotate(vec3 p, vec3 axis, float angle, sdf3d primitive){
    mat3 rot_matrix = get_rot_matrix(axis, -angle);
    return primitive(rot_matrix * p);
}

// Skaalaus kertoimella s
vec3 scale(vec3 p, vec3 s, sdf3d primitive){
    return primitive(p/s) * s;
}

```

Listaus 2. Kappaleiden asemointi avaruudessa. Kaavojen primitive-parametrin tilalle laitetaan kappaleen etäisyyskenttä. Translaatio siirtää etäisyyskentän kappaleita vähentämällä syötepisteestä halutun siirtomatkan. Kappaleen pyörittämistä varten syötepistettä pyöritetään vastakkaiseen suuntaan matriisin avulla. Pyöritysmatriisin määrittely on melko sotkuinen, joten se on piilotettu koodissa `get_rot_matrix`-funktion taakse. Kappaleen skaalaus tehdään jakamalla ja kertomalla skaalauskerroimella ennen ja jälkeen etäisyyden laskemisen.

```

float opUnion(float d1, float d2) {
    return min(d1,d2);
}

float opIntersection(float d1, float d2) {
    return max(d1,d2);
}

float opSubtraction(float d1, float d2) {
    return max(-d1,d2);
}

```

Listaus 3. Joukko-operaatioita etäisyyskentille. Syötteessä on kahden etäisyyskentän antamat arvot ja paluuarvona näiden etäisyyskenttien unioni, leikkaus tai vähennys. Unioni on eksakti operaatio, mutta kahdesta muusta tulee vääristymää kenttään. Kuvassa 3 nähdään, miltä nämä operaatiot näyttävät pallolle ja laatikolle.

matkaa pintaa kohti. Yliarviointi saattaa toisaalta johtaa siihen, että säde menee pinnasta läpi, mikä on harmillista algoritmin kannalta.

Vääristynyttäkin etäisyyskenttää voi käyttää askelluksessa turvallisesti, kunhan askelpituutta pienennetään sopivasti. Jos esimerkiksi kenttä on vääristynyt kolminkertaisesti, täytyy askelkoko jakaa vähintään kolmella, jotta säteenaskellus toimii taas oikein, joskin hitaammin kuin aikaisemmin. Muunnoksen vääristyskerroin voidaan laskea sen derivaatasta, mutta käytännössä riittää kokeilla käsin eri askelpituuksia, kunnes sopiva löytyy.

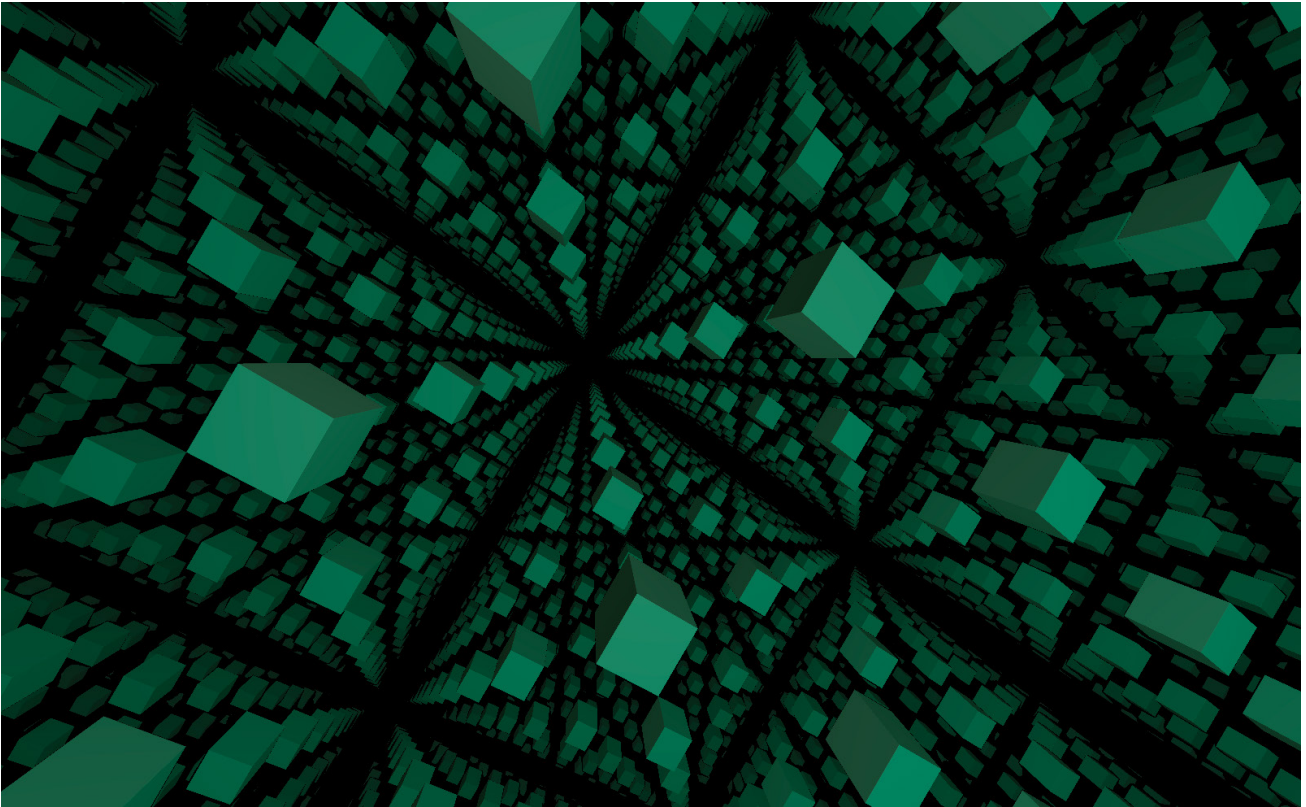
Yksi erityisesti pienissä introissa hyödyllinen muunnos on ottaa syötepisteestä komponentteittain modulo ennen sen syöttämistä etäisyyskenttään. Esimerkiksi jos x-koordinaatista otetaan modulo kymmenellä, maailma toistuu kymmenen yksikön välein x-akselin suunnassa. Kun sama tehdään myös y- ja z-suunnassa, maailma toistaa itseään joka suunnassa (kuva 5). Näin saadaan koko ruutu täytettyä tavaramalla vain pienellä määrällä koodia. Tämän kanssa tulee kuitenkin olla varovainen, koska tämä vääristää etäisyyskenttää, jos kappaleiden kopioiden pinnat ovat lähellä toisiaan.

Toteutus

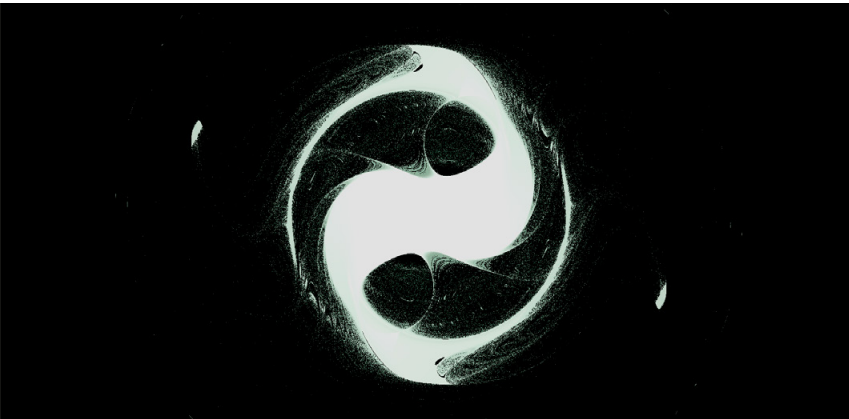
Lähes kaikki RMSDF-moottorin koodi voidaan kirjoittaa suoraan fragmentti-shaderiin eli GPU-ohjelmaan, joka lasketaan jokaiselle pikselille rinnakkain, mutta tämä vaatii hieman alustusta muualta. Aluksi tarvitaan koodia (esimerkiksi assembly- tai C-kielellä), joka linkittää tarvittavat kirjastot, lataa shaderit, avaa ikkunan ja kutsuu piirtofunktiota silmukasta.

Jotta kaikki tämä saadaan pieneen tilaan, ei ole varaa linkata edes C:n ajoaikakirjastoon, joka tulisi normaalisti automaattisesti mukaan. Tämän takia ohjelman suoritus ei alakaan main-funktiosta, vaan vähemmän tunnetusta `_start`-funktiosta, joka normaalisti kutsuisi main-funktiota. Koska mitään ei ole valmiina, alustukseen tarvitaan paljon matalan tason tietämystä käännösprosessista, dynaamisesta linkittämisestä ja käyttöjärjestelmän toiminnasta.

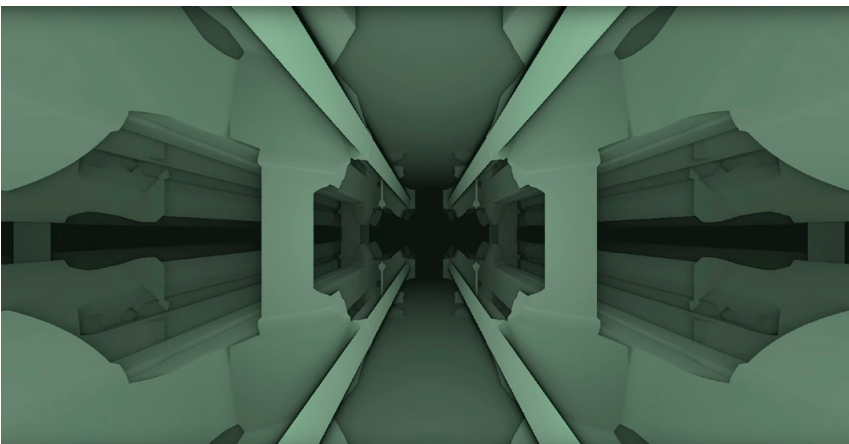
Onneksi tätä matalan tason hommaa ei tarvitse vuonna 2019 enää tehdä itse. Windowsille löytyy **Yzin** mainio työkalu `CompoFiller Studio` (kameli.net/compo-fillerstudio), joka paketoii shaderin



Kuva 5. Modulon ottaminen syöte pisteestä ennen etäisyyden laskemista toistaa kappaleita äärettömästi. Efekti on hyvä erityisesti pienissä introissa, koska sillä saadaan täytettyä koko ruutu pienellä määrällä koodia.



Kuva 6. Grafiikkakoodauksessa on se hauska piirre, että välillä saattaa törmätä odottamattoman hienoihin tuloksiin, koska ohjelmassa on bugi. Tässä ensimmäinen yritykseni donitsin kiertämiselle.



Kuva 7. Tech power ride by DigiMind – säteenaskellusta yhden kilotavun Windows-introssa kesän 2019 Assemblyillä. Teos sijoittui kisassa toiseksi.

ajettavaksi .exe-tiedostoksi. Mukaan saa myös musiikit 4klang-formaatissa. Työkalu pakkaa intron suositulla Crinkler-työkalulla.

Linuxilla työskentely on hieman hankalampaa. Yksi hyvä lähtökohta on Faemiyah-groupin dnload-työkalu (github.com/faemiyah/dnload), joka ottaa syötteenä C-koodin ja palauttaa minimoidun ELF-binääriin. Se hyvä puoli Linuxissa on, että normaalissa Ubuntu-asennuksessa tulee automaattisesti mukana laaja Unix-työkalupakki, joka sisältää esimerkiksi pakkaustyökalut zip ja xz. Hyvä tapa pakata Linux-intro on kirjoittaa se Bash-scriptiksi, joka purkaa ja ajaa itsensä. Sama on mahdollista myös macOS-ympäristössä.

Javascript ja nettiselaimet tarjoavat toisenlaisen tavan rakentaa intro. Selainto pääsee käsiksi näytönohjaimen WebGL:n kautta, ja suorituskyky yltää natiivien introjen tasolle, kunhan kaikki raskas laskenta tehdään näytönohjaimen puolella. Intro kirjoitetaan tyypillisesti pakatuksi png-tiedostoksi, jonka selain osaa purkaa ja sitten ajaa.

Loppusilaukset

Kuvatuilla tekniikoilla saa jo monenlaisia pikselikasoja lentämään ruudulla. Kun perusmoottori ja etäisyyskenttä on valmis, on aika lisätä erilaisia

```

float sdf(vec3 p){ // Donitsi
    vec3 offset = vec3(0.0, 1.3, -1.8);
    p.yz = vec2(p.y/sqrt(2.0)+p.z/sqrt(2.0),p.y/sqrt(2.0)-p.z/sqrt(2.0));
    p += offset;
    return length(vec2(length(p.xz)-1.0,p.y)) - 0.3;
}

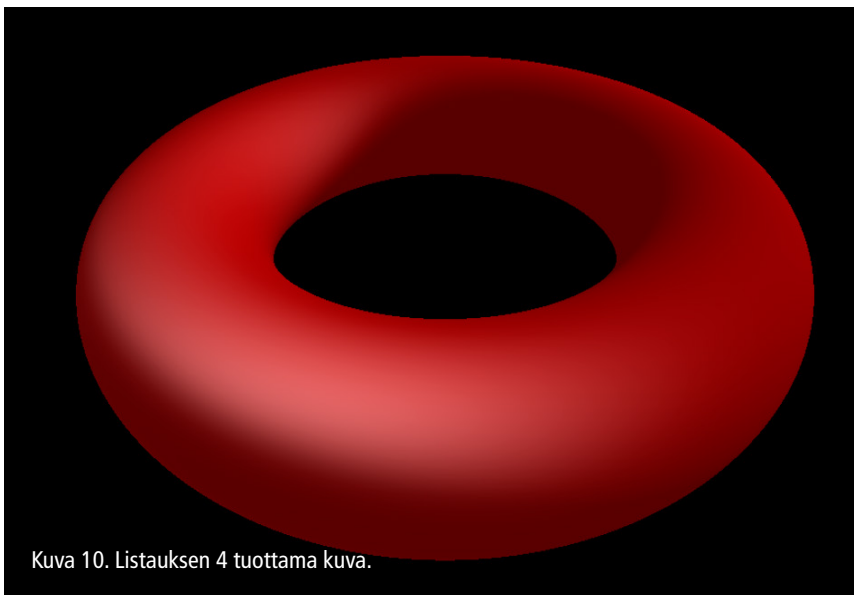
vec3 get_normal(vec3 pos){ // Numeerinen approksimaatio pinnan normaalille
    vec3 eps = vec3(0.01,0,0);
    return normalize(vec3(
        sdf(pos + eps.xyy) - sdf(pos - eps.xyy),
        sdf(pos + eps.yxy) - sdf(pos - eps.yxy),
        sdf(pos + eps.yyx) - sdf(pos - eps.yyx)
    ));
}

vec4 get_color(vec3 surface){ // Pikselin väri Phongin valaistusmallilla
    vec3 normal = get_normal(surface);
    vec3 l = vec3(-1,-1,-0.2); // Valon tulosuunta
    vec3 l_prime = l + 2.0*dot(l,normal)*normal;
    float cos_angle = max(dot(-l_prime, -surface),0.0)/(length(l_prime) * length(-surface));
    vec3 specular = vec3(1.0,1.0,1.0) * 0.8 * pow(cos_angle,2.5);
    vec3 diffuse = vec3(1.0,0.0,0.0) * 0.2 * max(dot(normal, -1),0.0);
    vec3 ambient = vec3(1.0,0.0,0.0) * 0.3;
    return vec4(clamp(diffuse + specular + ambient,0.0,1.0), 1.0);
}

void mainImage(out vec4 outputColor, in vec2 fragCoord){
    // Skaalataan pikseleiden koordinaatit välille [-1,1]
    vec2 pixel_pos = (fragCoord.xy / iResolution.xy - vec2(0.5,0.5)) * 2.0;
    pixel_pos.x *= 16.0/9.0; // Kuvasuhde
    // Luodaan säde kamerasta yhden yksikön päässä leijuva
    // kuvitteellisen ikkunan läpi annetun pikselin kohdalta
    vec3 ray = normalize(vec3(pixel_pos.x, pixel_pos.y,-1));
    // Tehdään 50 iteraatiota säteenaskellusta etäisyyskentän avulla
    vec3 pos = vec3(0.0,0.0,0.0);
    for(int step = 0; step < 50; step++) pos += ray * sdf(pos);
    if(sdf(pos) > 0.01) // Ei osumaa
        outputColor = vec4(0,0,0,1);
    else // Osuma
        outputColor = get_color(pos);
}

```

Listaus 4. Säteenaskellusmoottorin koodi. Koodin mainImage-funktio ottaa parametrina ruudun pikselin koordinaatit ja palauttaa pikselille lasketun värin. Kamera on origossa ja katsoo pitkin negatiivista z-akselia. sdf-funktio määrittelee donitsin kameran eteen. Moottorilla voi leikkiä Shadertoyssa osoitteessa shadertoy.com/view/wlSSW1.



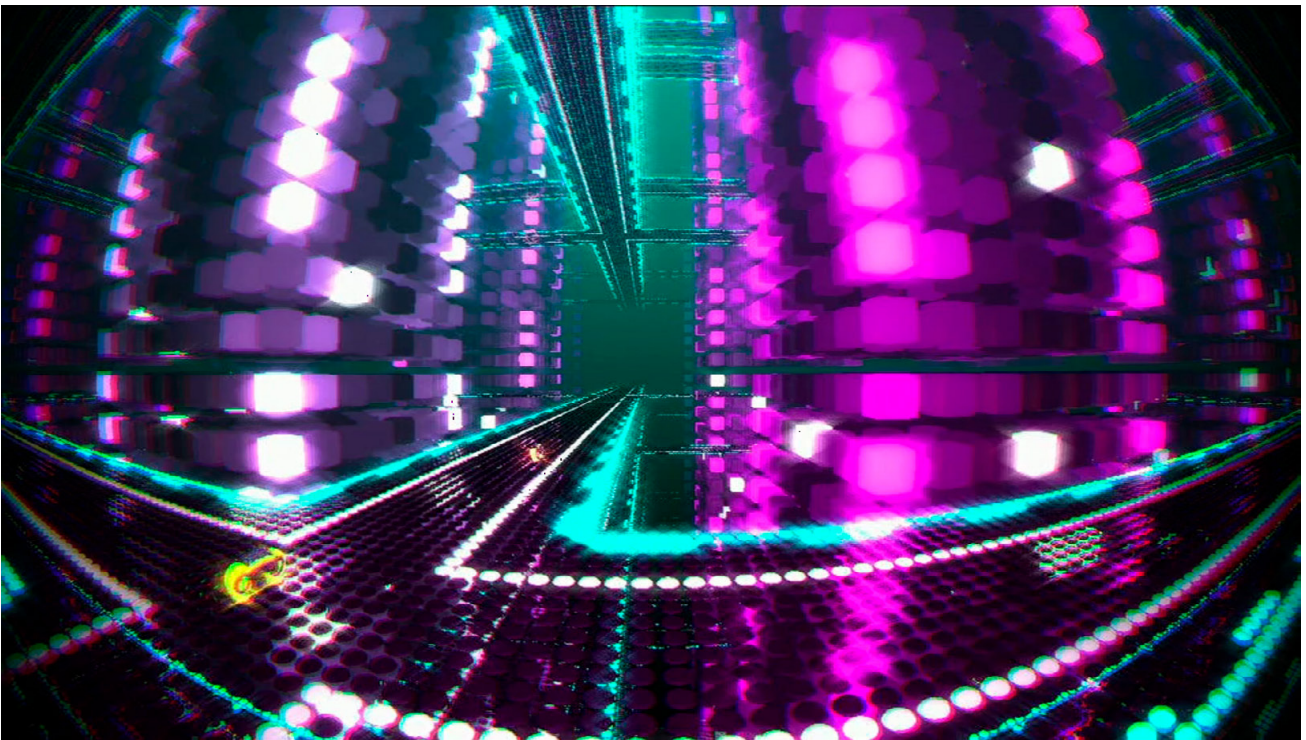
Kuva 10. Listauksen 4 tuottama kuva.

visuaalisia efektejä niin paljon, kuin kokorajaan mahtuu. Esimerkiksi varjojen lisääminen tuo jo paljon lisää realismia. Varjoja voi lisätä askeltamalla pinnalta kohti valonlähdettä ja katsoamalla törmääkö se johonkin pintaan ennen valonlähdettä. Tässä tapauksessa pikselin kirkkautta pienennetään.

Kaikenlainen kuvan jälkikäsitteily on myös yleensä kotiin päin. Inspiraatiota efekteihin voi hakea esimerkiksi katsomalla Youtubesta edellisvuosien teoksia ja selailemalla muiden tekemiä shadereita shadertoy.com-sivustolla. Koodiriveissä erityisen halpoja efektejä saa aikaan, kun sotkee moottorin parametreja väliaikaisesti. Kannattaa kokeilla, saisiko kuvan rikottua ja korjattua näyttävän näköisesti. Esimerkik-



Kuva 8. Luminainen neljän kilotavun Linux-intro Mount Torii vuoden 2016 Assemblyiltä. Intro käyttää RMSDF-moottoria useammalla valonlähteellä. Teos sijoittui kisassa toiseksi.



Kuva 9. Highway 4k by ½-bit Cheese & Tekotuotanto. Tämä säteenaskellettu selain-intro voitti vuoden 2013 Assemblyjen 4k-kisan. Intron lähdekoodi selityksineen löytyy osoitteesta yilammi.com/webgl/highway4k/Making%20of%20Highway%204k.pdf. Tekstin mukaan intron tekemiseen kului noin 300 tuntia aikaa, eli noin kaksi kuukautta täysipäiväistä työtä.

si hetkellinen ylivalotus voi toimia oikeassa yhteydessä hyvin. Efekti-ideoita voi prototyypailla helposti Shader-toy-sivustolla.

RMSDF on siitä mielenkiintoinen tekniikka, että se ei ole levinnyt juuri demoscenen ulkopuolelle. Syynä lie-nee moottorin heuristinen luonne: säteet eivät välttämättä pääse koskaan

maaliinsa, jos matkalla on paljon lähellä olevia pintoja. Introa tehdessä nämä ongelmat voidaan yleensä kiertää muokkaamalla maailmaa tai kuvakulmaa sopivammaksi, mutta esimerkiksi videopelimoottorissa piirtoalgoritmin olisi suotavaa toimia oikein mistä tahansa kuvakulmasta. Varsinkin introissa käytetyt vääristävät muunnokset

ovat oikeasti summittaista hakkerointia, joka tuunataan toimivaksi valitusta kamerakulmasta parametreja käsin säätäen. Tiivistäen: introssa maailma palvelee moottoria, mutta yleiskäyttöisessä koodissa moottori palvelee maailmaa. 🤖

Lehden tekijät

Edesmennyt pomoni opetti minulle, kuinka tietokonelehteä ei pidä tehdä. Luonnollisesti päätin tehdä juuri niin.

Janne Sirén

Olen lähes koko aikuisikäni ollut oma herrani ja narrini – yrittäjänrenttu kun olen – eikä minulla ole koskaan ollut täyspäiväisesti pomoa. Mutta sen hetken, kun minulla oli herra osa-aikaisesti, herran nimi oli **Markku Alanen**. Paikka oli tietokonelehti MikroBitti vuosina 1995–2002, keskellä Markun kymmenvuotista päätoimittajapestiä. Markku ja tytär Ninnu kirjoittivat itse Markun MikroBitin tarinan sivuilla 12–15 paremmin kuin osaisin. Mutta kerron nyt kolme asiaa, jotka opin Markulta siitä, mikä tekee hyvän lehden.

Tarkalleen ottaen Markku oli pomoni pommo. Lähin esimieheni oli MikroBitin toimituspäällikkö **Pasi Andrejeff**, joka ainoana bittiläisenä on selvinnyt lehdessä läpi kaikkien mulistusten, kutistusten ja omistajanvaihdosten noista ajoista tähän päivään. Pasi on hieno mies, enkä sano tätä vain siksi, että tämän suorituksen perusteella Pasiin täytyy tietää, missä luurangot ovat. Hieno mies. Mutta tarinani MikroBitissä ei alkanut Pasista, vaan se alkoi tästä toisesta hienosta miehestä.

Tarinani Bitissä alkoi, kun Markku uhkasi haastaa minut oikeuteen.

Läksy 1

Olin 1990-luvulla aktiivinen Amiga-tietokoneen harrastaja ja perustamassa Suomen Amiga-käyttäjät ry:tä vuosina 1993–1994. Samoihin aikoihin MikroBitti siirtyi vaihteittain Markun hallintaan ja uudistui. Yksi uudistus oli Amiga-tietokonetta koskevan ja alati kutistuvan asian karsinoinen lehteen erillisille Amiga-apartheid-sivuille. Toinen oli lehden tilaajille suunnatun MBnet-purkin perustaminen. Nämä tapahtumat toivat minut ja Markun ensimmäisen kerran (napit) vastakkain.



Kävipä nimittäin niin, että eräs MikroBitin silloinen avustaja kirjoitti *sfnet.atk.amiga*-uutisryhmässä varmaa huhua MikroBitin Amiga-sivujen lähestyvistä lopettamisesta tyystin – ja menin kirjoittamaan tämän tiedon edelleen uudenkarheaan MBnetiin. En tähän päivään mennessäkään tiedä oliko huhussa tuolloin jotain perää, mutta sen tiedän, että MBnet-viestiini reagoitiin korkealta taholta: Markku haastoi viestin ja uhkasi haasteella liiketoiminnan sabotoimisesta.

Tuskinpa Markku ihan tosissaan oli. Minäkin olin vilpiton, sillä MikroBitin avustaja todella oli kirjoittanut näin uutisryhmään ja olin uskonut häntä. Markku ei ollut tiennyt asiasta. Tämä keskustelu ja MikroBitin päätös rajata Amiga omaksi osastokseen olivat samalla ensimmäinen ja viimeinen asia, josta muistan olleeni Markun kanssa eri mieltä.¹

Sen sijaan, että olisin joutunut mustalle listalle pään aukomisesta, Mar-

kun MikroBitti tarjosi minulle tämän jälkeen töitä. Ilmiö toistui muidenkin kohdalla. Markku antoi ennakkoloukkomasti äänen heille, joilla oli sanottavaa. Ajatus lehden tekemisestä yhteistyössä lukijoiden kanssa ei siis ollut sanahelinää, mutta mikä tärkeintä, ääneen päästettiin myös erimieliset.

En muista, että Markun MikroBitissä minua olisi kertaakaan ojennettu siitä, mitä saa sanoa tai mistä saa kirjoittaa. Jopa Skrollissa on ojennettu useammin.

Läksy 2

Jo ennen Markun päätoimittajapestiä, 1990-luvun alkupuoliskolla, MikroBitti oli tosiharrastajien mielestä mennyt pilalle: ohjelmalistausten painaminen ja Bitti-leirit oli lopetettu, kotitietokoneet oli karsittu minimiin harmaan toimistoraudan tieltä ja tee-se-itse-mennon korvasi yhä useammin valmisohjelmavertailut.

MikroBitti peilasi tässä toki aikaansa. Kotitietokoneet väistyivät henkilökohtaisten tietokoneiden tieltä kaikkialla ja yhä harvempi tarvitsi ylitioteknisiä taitoja käyttääkseen niitä. Mutta oli kriittisissä peräinkin – siihen on syynsä, miksi Skrollin vuonna 2012 käynnistänyt Retrobitti.jpeg-meemi (ks. Skrolli 2015.1)

¹ Karsinoinnin sijaan Amiga-, Mac- ja muut järjestelmät olisi voitu huomioida MikroBitissä Skrollin tapaan kautta lehden, pienellä artikkelitekijöiden yhteistyöllä. Esimerkiksi tusinan sivun PC-tulostinartikkelista saa parilla sivulauseella tai laatikolla myös Amiga- ja Mac-jutun. Tämä kulttuuriteko olisi opettanut ysärin PC-jonnapolvelle karsinoita enemmän tietotekniikan moninaisuudesta.

mallasi vanhempaa 1980-luvun Bittiä. Jotain tee-se-itse-kulttuurista oli lipsussa Bitin ulottumattomiin, juuri kun Markku astui laivan isoimpaan ruoriin.

Tässä valossa ymmärsin vasta myöhemmin, kuinka arvokas asia Markun MikroBitin loppuvuodesta 1994 lanseeraama MBnet-purkki olikaan juuri tee-se-itse-kulttuurille. Siinä missä yksi suomalainen ATK-sukupolven puolikas oli kasvanut vaihtaen ohjelmalistauksiaan, kirjeitään ja ilmoituksiaan 1980-luvun MikroBitin sivuilla, toinen teki sen vuosikymmentä myöhemmin MBnetin linjoilla. Moni linjoilla tapahtuneista kohtaamisista jatkui lehden sivuilla – kuten minunkin kohdallani.

MBnetin vaikutus suomalaiselle indie-ohjelmistokehitykselle ja ohjelmointitaitojen kehittymiselle 1990-luvulla olisi akateemisen tutkimuksen arvoinen. En ihmettelisi, vaikka ysäriin PC-nuorisolle MBnetin vaikutus olisi absoluuttisesti suurempi, kuin lehden painettujen ohjelmalistausten ja Bitti-leirien merkitys kasarini kotitietokoneuorille.

Olemme huomanneet toimivan ekosysteemin merkityksen Skrollissa: Jokainen onnistunut numeromme, digitempauksemme tai tapahtumamme ruokkii lisää artikkeleita ja rohkaisee uusia kirjoittajia – mutta tätä ei tapahdu, jos tekemisemme on yleiselle yhä edelleen.

Läksy 3

Olin jo väistynyt MikroBitin leivistä opiskelujeni tullessa päätökseen 2000-luvun alussa, kun kuulin pikukulinnuulta julkaisijan suunnitelmista MikroBitille: Lehdestä olisi tulossa uuden tekniikan Tekniikan Maa (TM)... MikroBitistä tulisi MB. Tämä oli kauan ennen kuin MikroBitin kanteen viimein ilmestyi MB vuonna 2009 tai kun lehden nimi virallisesti vaihtui mB:ksi vuonna 2012, josta Markku kirjoitti Skrollin kolumnissaan Bitin kuolemana. Täystuho oli työn alla pitkään.

Hidas muutos MB:ksi alkoi jo vuonna 2004 Markun lähden aikoihin, kun entinen tietokonelehti siirtyi käsittelemään yhä enemmän kuluttajaelektronikkaa. Seuraava merkkipaalu oli viihde-elektronikkaan keskittyneen Hifi-lehden liittäminen MikroBittiin vuonna 2006. Odotetusti Hifi-lehden tekijät ja lukijat protestoivat. Hifin raunioille syntyi osin samoin voimin

kilpaileva Hifimaailma-lehti, joka on edelleen olemassa, toisin kuin MikroBitin Hifi-sivut.

Lehtifuusiot ovat vähän kuin verotusdynaamisia. Veroja nostettaessa voidaan toivoa, että verotulot kasvavat. Lehtiä yhdistäessä voidaan toivoa, että kahden lehden lukijat saadaan jokuinkin säilytettyä yhden lehden tuotantokustannuksilla. Tosiasiassa muutoksella on kuitenkin aina tapana vaikuttaa myös maksajapuolen käytökseen, minkä seurauksia voi olla vaikea (tai helppo...) ennustaa.

Monesti lehtifuusio onkin merkki lopunaikojen epätoivosta. Kun klassinen brittiläinen Sinclair-pelilehti Crash yhdistettiin Sinclair User -lehden 1990-luvulla, lopputuloksesta seurasi lähinnä vuosikymmenten vitsi. Hieno tuote vedettiin lokaan yhden kurjan lisävuoden tähden. Vain pavunlaskijat osaavat arvioida, myydytkö lehtiä tällaisessa tilanteessa enemmän kuin ilman fuusiota olisi myyty, mutta vuosikymmenten mainehaitta on ilmeinen maallikollekin.

Markku oivalsikin kolumnissaan (s. 14) jotain olennaista, kun hän puhuu sisältölähtöisyydestä ja lupauksesta siitä, mitä lehti on. Asiakas haluaa ostaa lupauksen mukaista sisältöä. Nähdäkseni Markku puolusti MikroBitin sisältölupauksia niin pitkään kuin pystyi. Markun pakkolähtö oli lehden tekijöille kuin sirpalekranasta toimituksen ovesta sisään, mutta ennen kaikkea se oli hitaasti vaikuttava ydinlaskeuma lehden sisällölle – ja lupaukselle. Jälkimmäinen oli (lähes) fataalia.

1+1=?

Pohdin samaa kysymystä joitakin vuosia sitten, kun nyt yli 25-vuotiaana, Saku-nimistä Amiga-e-lehteä aikanaan julkaisseen Suomen Amiga-käyttäjät ry:n eli ”Sakun” jatkosta keskusteltiin. Amiga-kerholle on 2010-luvulla vähemmän kysyntää kuin 1990-luvulla, ja uusia ideoita pyöriteltiin. Kerhoa on toki venytetty kevyesti Commodore-klubin suuntaan, mutta leikittelimme ajatuksella laajentumisesta yleisemmin vaihtoehtoisrautaan – tyyliin Suomen Alt-käyttäjät. Ajatus haudattiin onneksi nopeasti. On pysyttävä lupauksessa. On parempi kiinnostaa harvaa, kuin ei ketään. Saku porskuttaa edelleen.

Ei toimintojen yhdistäminen silti aina ole pahasta. Kun liiketoimintakauppojen jälkeen partakoneen käyt-

täjien elämäntapalehdiksi kuihtunut mB ja vanha MikroPC-lehti yhdistyivät uudeksi-vanhaksi MikroBitiksi vuonna 2015, veikkaan, että osien summa oli enemmän kuin molemmat erikseen. Samassa toivossa teimme tänä kesänä Skrollinkin toimituksessa rohkean päätöksen: toivotamme ensi vuonna pelilehti *Retro Rewindin* tervetulleeksi Skrollin kansien väliin. Mikä voisi mennä vikaan...?

Olenaisena pontimena on, että uudistuksemme tuntuu sisällöllisesti sekä kiinnostavalta monille lukijoistamme että kulttuurihistoriallisesti arvokkaalta. Uskomme myös voivamme toteuttaa uudistuksen ottamatta kummastakaan lehdestä mitään pois. Skrollia ja Retro Rewindia ilmestyy ensi vuonna vuotuisissa sivuissa arviolta yhtä paljon kuin aiemminkin, kumpikin erikseen omalle asialleen uskollisesti toimitettuna. Lehdet ainoastaan taitetaan samoihin fyysisiin kuoriin, mikä tekee niiden tarjoamisesta pienessä maassa hieman helpompaa.

Toivon, että olemme kuunnelleet Skrolli-yhteisöä – ja Markkua – tarpeeksi.

Post Scriptum

Siirryin tämän Skrollin myötä lehden toimituspäälliköstä päätoimittajaksi. Viimeinen kerta, kun kuulin Markusta (välikäden kautta) oli elokuussa 2018. Markku oli todennut, että ”Janne tekee varmasti koko [Skrollin], jos sen päästää vapaaksi.” Kiitos luottamuksesta tahoillanne, Skrolli-porukka ja Pomo.

Toivon, että en joudu tekemään Skrollia yksin vastaisuudessaan, mutta Skrollin kirjoitetun historian mukaan (ks. Skrolli 2018.1) jokaisella päätoimittajalla on lähestulkoon ”oltava” oma ajatus Skrollin kaltaisesta paperilehdestä. Näitä ovat olleet **Ville-Matiaksen Tritti**, **Mikon Kunnan Lehti** ja **Tapion Osaaja**.

Voin nyt paljastaa omani vuoden, parin takaa – vakuutukseni siltä varalta, että Skrolli uudistuu partakonekulttuurin peruslehti sK:ksi. Minun paperilehteni nimi olisi *Saku – Suomen tietokonekäyttäjien lehti*. Saku ry:n puheenjohtaja **Anu Seilonen**, soitan, jos varasuunnitelmaa tarvitaan. Tosin myös Saku lyhenee sK:ksi... 🌱

Vanhat Skrollit maksuttomina pdf-lehtinä sekä Saku-e-lehdet verkkojatkoillamme: skrolli.fi/numerot



Minitel-standardin mukainen päätte koostui näytöstä, näppäimistöstä ja modeemista. Ensimmäisissä laitteissa ei ollut lainkaan käyttömuistia, mutta standardin kehittyessä niihin lisättiin 30 vuoden aikana värejä, värisävyjä, muistia ja toiminnallisuutta. Ne eivät kuitenkaan olleet missään vaiheessa tietokoneita, joissa voitaisiin ajaa ohjelmia. Kuva: A1AA1A.

landin ja Kevin Driscollin Minitel – Welcome to the Internet -teos ranskalaisesta Internetin edeltäjästä lankeakin herkulliseen maaperään. Se on näet ensimmäinen englanninkielinen akateeminen tutkimus Minitelistä ja upea sukellus sen digitaaliseen maailmanhistoriaan ja kulttuuriin.

Valtiollinen haaste

Minitel ei ollut Internet eikä varsinaisesti sen kilpailijakaan, vaikka palvelut olivat olemassa myös yhtäaikaan. Molemmat saivat alkunsa valtiohallan projekteina, kylmän sodan keskellä.

Ranskassa kylmä sota johti 1960-luvulla vastareaktioon, oman kulttuurin ja merkityksen alleviivaamiseen. Maa rakensi sekä atomipommin että Silicon Valley -tyyppisen teknologiakeskittymän Bretagnen maakuntaan. Yksi siellä syntyneistä innovaatioista oli vuonna 1974 kehitetty laitteistoriippumaton Antiope-standardi, joka määritteli kuinka informaatiota voidaan näyttää videonäytöillä. Vuonna 1978 puolestaan ideoitiin Antiopea käytävä puhelinmodeemeilla käytettävä tietopalvelu. Se oli myös oiva tekosyy Ranskan rappeutuneiden tietoverkkojen kunnostamiseen. Parhaimmassa tapauksessa palvelun tuotot rahoittaisivat remontin.

Halpaa tai nopeaa se ei silti ollut. Visio oli niin valtava, ettei sitä voitu toteuttaa kvartaalirahoituksella, vaan ruoriin tarvittiin suunnitelmaa pitkäjänteisesti toteuttava valtiovalta. Minitelillä ei myöskään ollut esikuvia, joten kaikki rakennettiin tyhjästä. Palvelu täytyi survoa kansalaisten kurkuista alas luomalla kysyntä keinotekoisesti. Yksi tapa oli tarjota kiinnostavaa sisältöä ja palveluja. Toinen oli jakaa Minitel-päätelaitteita ilmaiseksi. Valtiollisessa mitassa tilattujen Minitelien valmistaminen oli myös valtava potku Ranskan teknologiateollisuudelle.

Innovaatiorypäle

Minitel oli keskitetty järjestelmä, jossa kaikki liikenne kulki valtiollisten reitittimien kautta. Palvelu sisälsi alusta

Minitel – ranskalainen esinetti

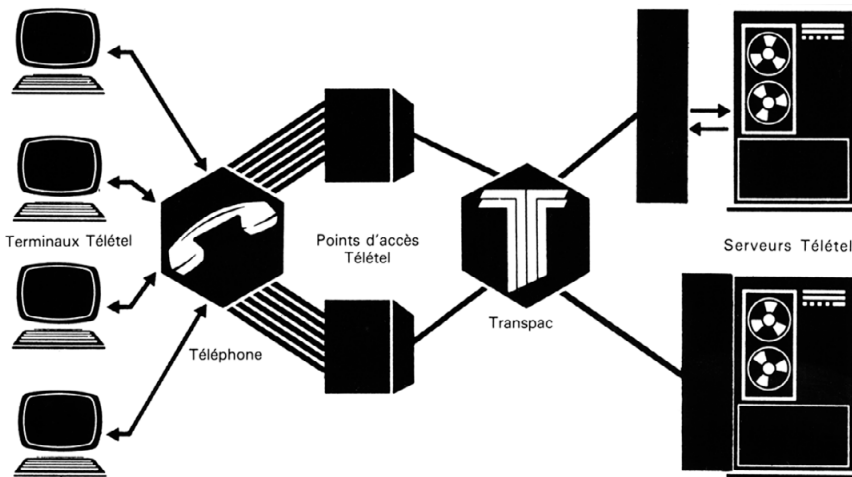
Teksti: Jukka O. Kauppinen

Kuvat: Wikimedia Commons (ks. kuvatekstit), MIT Press

Ranskalainen Minitel-palvelu nosti maan sähköisten palvelujen eturiviin 1980-luvulla, kauan ennen Internetiä. Minitelin todellinen laajuus on kuitenkin ollut mysteeri etenkin kielimuurin takia. Uusi englanninkielinen teos avaa sen historiaa, tekniikkaa ja kulttuuria kattavasti.

Ranskassa nähtiin 1970-luvun lopulla seuraavaa kvartaalia pidemmälle. Nokkelat insinöörit visioivat puhelinmodeemien kautta toimivan järjestelmän, joka voisi yhdistää koko Ranskan yhteisen digitaalisen palvelun ääreen. Valtiovalta tarttui ehdotukseen ja loi valtiollisen posti- ja lentäinlaitoksen infrastruktuuriin pohjautuvan Téletel-verkkopalvelun, jota käytettiin Minitel-päätteillä, kuin kaksisuuntaista teksti-tv:tä. Sittemmin palvelu tunnettiin yksinomaan Minitelinä.

Minitelin olemassaolo tiedettiin Suomessa ja muuallakin maailmassa, mutta koska Ranska on Ranska, niin sen mittakaava ja etenkin sisältö ovat olleet kovin vaikeasti hahmotettavissa. MIT Pressin julkaisema, **Julien Mail-**



Remarque importante pour les serveurs : un même accès physique au réseau Transpac peut acheminer à la fois du trafic Télétel et du trafic téléinformatique.

Minitel – Welcome to the Internet

Kirjoittajat:

Julien Mailland ja Kevin Driscoll

Kustantaja: MIT Press

236 sivua

E-kirja, kovakantinen, pookari

Hinta: Noin 17 euroa (Kindle, Nook Book), 17–28 euroa kirjana (Amazon)

Lisätietoja:

mitpress.mit.edu/books/minitel

Minitel-käyttäjät ottivat modeemiyhteyden palvelunumeroon, josta yhteys reititettiin joko Télételin omille tai kaupallisten operaattorien palvelimille. Myöhemmin Télételiin rakennettiin silta-ksia myös muiden operaattorien verkkopalveluihin. Kuva: Orange/DGCI.

saakka keskitetyn maksujärjestelmän, jossa maksullisten palvelujen veloitus lisättiin puhelinlaskuun. Minitel tarjosi valmiin infrastruktuurin ja laskutuspalvelun lisäksi avoimet rajapinnat, joihin sisällöntuottajat pystyivät kytkeemään omat palvelimensa käyttöjärjestelmästä ja arkkitehtuurista riippumatta. Tarvitsi vain keksiä hyvä idea, ohjelmoida se ja kytkeä verkko. Syntyi täysin uudenlainen it-teollisuus riskinottajineen, innovaatioineen ja menestyjineen.

Käyttäjää houkuteltiin ilmaisilla palveluilla, joista alkuun tärkein oli sähköinen puhelinluettelo. Muna ja kana syntyivät yhtäaikaan, ja käyttäjät lipuivat

ilmaispalveluista myös kaupallisten, minuuttitaksalla laskutettujen sisältöjen ääreen. Ranska tarttui Miniteliinsä kaksin käsin, ja pian sen uumenissa tialttiin juna-, elokuva- ja teatterilippuja, asioitiin verkkokaupoissa, katsottiin säätiedotuksia ja pulistiin keskustelupalstoilla. Kaikkea tätä jo 1980-luvulla, ennen Internetiä. Minitel-päätteitä sijoitettiin myös työpaikoille ja julkisiin tiloihin.

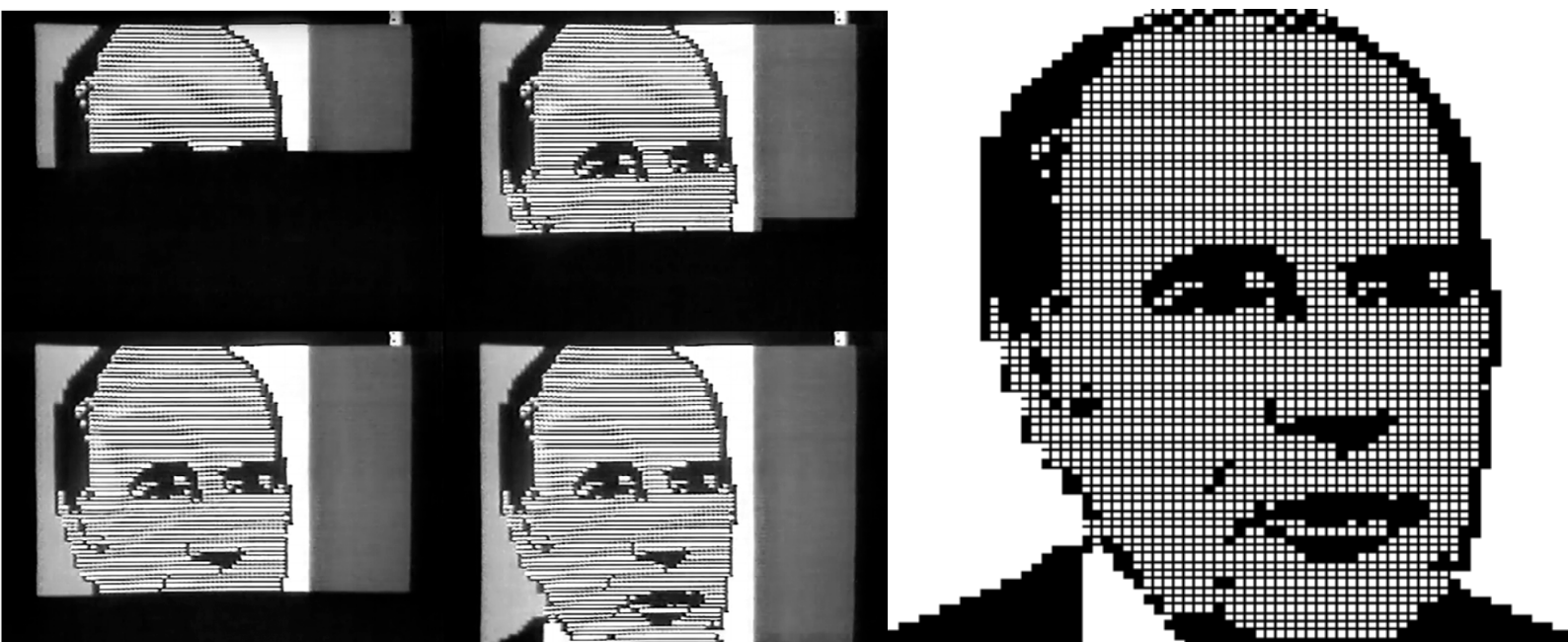
Valtiovaltaa pidetään usein byrokrattisena ja ymmärtämättömänä organisaationa. Minitel oli kuitenkin valtiollinen rohkea ja edistyksellinen loikka uudenlaiseen digitaaliseen maailmaan aikana, jolloin kotitietokoneet-

kaan eivät olleet vielä arkipäivää. Se loi maahan elävän verkkokulttuurin ja yhdisti maan yhdeksi digitaaliseksi pulinaboksiksi.

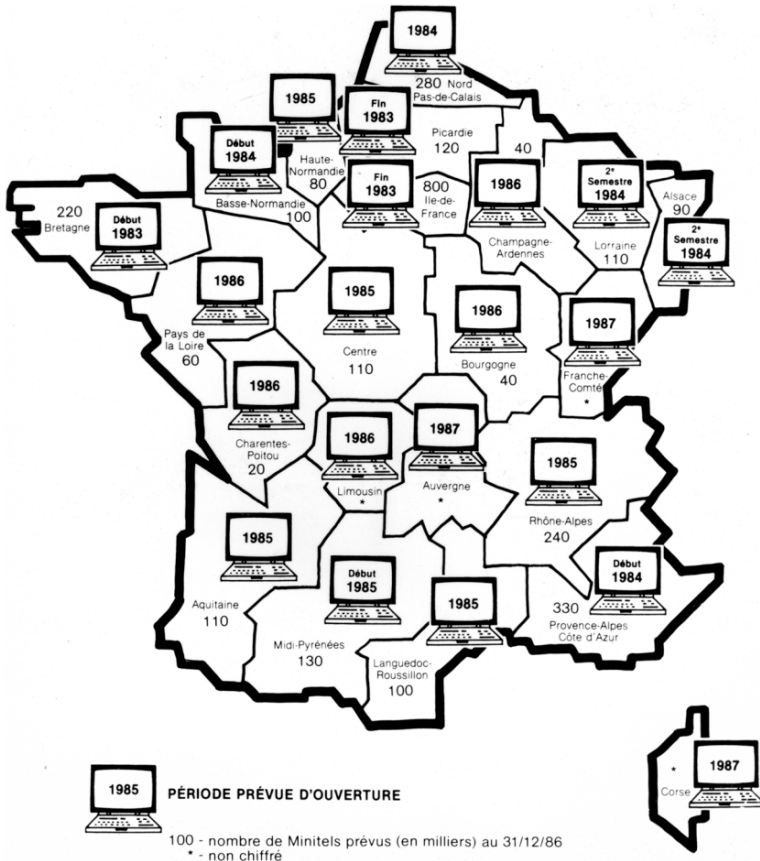
Tuhmat jutut modernisoivat Ranskan telen

Valtiollinen ohjaus toi toki omat pulmansa. Sisältökomitea päätti, mitä palveluja Miniteliin sai liittää, ja uuden tulokkaan hyväksyntä saattoi kestää kuukausia. Poliitikot käyttivät Miniteliä keppihevosenä omien agendojensa ajamiseen, eräs jopa pelkäsi sen johtavan joukkopidätyksiin Neuvostoliiton miehittäessä Ranskan.

Skrolli-kulmasta nähtynä riemasuttavinta lienee Minitelin inspiroima digitaalinen kulttuuri. Hakkerit modasivat päätteitä, rakensivat omia Mi-



Vuoden 1981 presidentinvaalien tulos paljastettiin televisiossa näyttämällä, kuinka Minitel-päätte piirsi ruudulle merkkigrafiikkakuvan maan uudesta presidentistä. Kuva: INA.

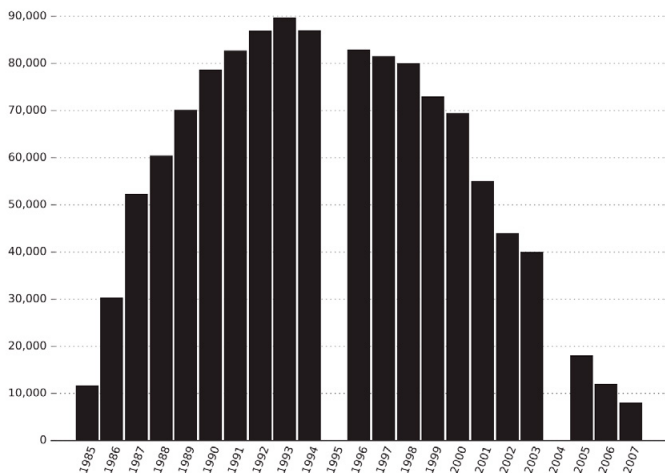


Minitelin ympärille kasvoi rikas yhteisö ja kulttuuri, aina omia aikakauslehtiään myöten. Ranskalainen pop-yhtye Liaison julkaisi vuonna 1987 *Minitel...Miniquoi?* -singlenkin.

rasta, joten Ranskan teleministerikin totesi: ”Ei meillä ole varaa sulkea niitä.” Voinee sanoa, että pinkillä sisällöllä rakennettiin uusiksi Ranskan puhelinverkosto.

Vaan eihän Minitel ikuinen ollut. Ihmisten tavat viestimiseen muuttivat, eikä 1970-luvulla suunnitellun järjestelmän tekniikkaa pystytty uudistamaan tarpeeksi haastamaan Internetin nopeampia yhteyksiä ja värikkäitä verkkosivuja. Minitel oli kuitenkin yhdistänyt Ranskan ja opettanut useamman sukupolven viestimään digitaalisesti. Valtiovallan ohjaus ja tuki sekä järjestelmän hyvät tekniset ratkaisut tekivät siitä vuosikymmenien mittaisen menestystarinan, jossa riittää kerrottavaa enemmänkin kuin kirjan 236:een saati tähän Skrollin kolmeen sivuun mahtuu. Lämmin suositus tälle teokselle, joka käsittelee Miniteliä hienosti ja kattavasti monilta eri kulmilta, kritiikkiäkään unohtamatta. 📖

Minitelien yleistymistähti. Vuonna 1983 Bretagnessa jaettiin 220 000 ilmaista Minitel-päätettä. Kuva: Orange/DGCI.



Minitel oli laajimmillaan vuonna 1993. Tällöin käytössä oli 6,5 miljoonaa Minitel-päätettä, ja 11 miljoonaa ranskalaista käytti palvelua lähes 90 miljoonaa tuntia. Kuva: Orange/DGCI.

nitel-verkkojaan ja loivat merkkigrافیikkataidetta. Voimallisinta sisältöä olivat *les messageries* -keskustelupalstat, joissa on yhtäläisyyksiä BBS-purkien keskusteluihin. Messengeriet olivat kuitenkin koko kansakunnan käytettävissä: kuka tahansa pystyi olemaan niihin yhteydessä mistä tahansa. Suosituimmilla maksullisilla palstoilla oli satojatuhansia käyttäjiä, eikä kukaan yllätynne siitä, että ne olivat seksuaalisävytteisiä tai suoraa pornoa. Myös seksuaaliset vähemmistöt ja muut

marginaaliryhmät löysivät niistä turvallisia kokoontumispaikkoja.

Tämä johti palvelujen omien moderaattorien ja valtiollisten sensorien vuosikymmeniä jatkuneeseen tasapainoiluun, joissa punnittiin yhteiskunnallisten normien ja vapaan sanan rajoja. Olkoon Ranskalle kunniaksi se, että tämä Pinkki Minitel miellettiin kuitenkin normaaliksi, jopa tärkeäksi osaksi yhteiskuntaa.

Pinkeistä messaregieista muuten roipisi 80 prosenttia Minitelin frangivir-

TELESAMPO – suomalainen vastine

Myös Suomessa havaittiin Minitelin suosio ja merkitys. Tele avasi kotimikrojen kautta modeemilla toimivan videotex-palvelu Telesammon vuonna 1984. Telesammolla ja HPY:n vastaavalla Infotelillä oli huippuvuotensa noin 350 000 käyttäjää, mutta pienellä liekillä ylläpidetyt palvelut katosivat hetkessä Internetin yleistessä.



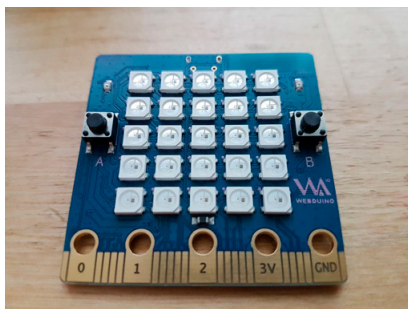
BPI-Bit, ESP32 ja MicroPython

Markkinoilla on useampiakin opetuskäyttöön tarkoitettuja kortteja, joiden avulla on helppo päästä alkuun mikrokontrollerien ohjelmoinnissa. Tässä jutussa kerrotaan kiinalaisesta BPI-Bitistä, joka edullisesta hinnastaan huolimatta sisältää monipuolisia toimintoja.

Teksti: Jarno Lehtinen Kuvat: Mitol Meerna, Jarno Lehtinen

Vuonna 2016 Englannissa julkaistiin opetuskäyttöön BBC Micro Bit -niminen kehitysalusta. Se on ARM-pohjaisen mikrokontrollerin sisältävä noin 5×4 cm:n kokoinen piirilevy, johon on integroitu muun muassa Bluetooth, painonappeja, antureita ja 5×5 ledin matriisi. Yksi sen ydinajatuksista on laskea kynnystä mikrokontrollerien ohjelmoinnin aloittamiseen. Micro Bitin lisäksi tarvitaan vain tietokone tai mobiililaitte sekä MicroUSB-johto, jonka kautta laite saa myös virtansa. Micro Bitin ARM-suoritin on laskentatehoiltaan melko vaatimaton, mutta toisaalta se ei vie paljon sähköä.

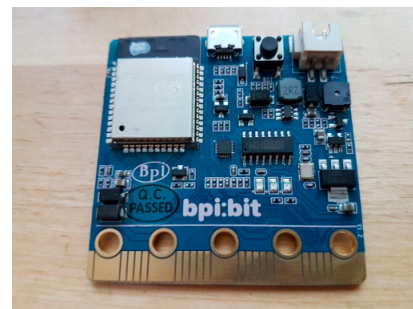
Kiinalaiset vastasivat Micro Bitiin julkaisemalla vuonna 2018 hyvin samankaltaisen kehitysalustan nimeltään BPI-Bit (myydään myös nimellä Banana Pi Bit). Se on kuin Micro Bit rajusti tehostettuna. Siinä on tehokas kaksiytiminen ESP32-mikrokontrolleri, joka sisältää Bluetoothin lisäksi myös wifin ja huomattavasti enemmän muistia sekä analogisia sisäänmenoja. Kehitysalustassa on kiihtyvyysanturien lisäksi yhdistetty asento-, kiihtyvyys- ja kompassianturi, lämpötila-anturi, 4 Mt flash-muistia, pieni summeri ja kaksi valoanturia. Lisäksi 5×5 ledin matriisi muodostuu yksiväristen ledien sijaan säädettävistä



RGB-LED-istä. BPI-Bit maksaa AliExpressissä postikuluneen tämän hetken kurssilla hitusen yli 20 euroa eli juuri sopivasti alle 22,5 euron rajan, joten siitä ei tarvitse maksaa arvonlisäveroa tuonin yhteydessä.

Kuten on Raspberry Pi'n ja sen luokusten kiinalaisten kilpailijoiden tapauksessa, rauta on huomattavasti esikuvaa järeämpää, mutta ohjelmistotuki ja dokumentointi ovat melko heikoilla kantimilla. BBC Micro Bitille löytyy runsaasti valmiita ohjeita ja ohjelmia, mutta BPI-Bitin käyttö vaatii enemmän tai vähemmän itse tekemistä ja toisinaan huonosti toimivien ohjelmien käyttämistä.

Tässä jutussa emme käytä erityisesti BPI-Bitiä varten tehtyjä ohjelmia, kuten Webduino Blocklya. Sen sijaan ohjelmoimme BPI-Bitiä kuten mitä tahansa ESP32:ta, johon on liitetty muita



laitteita. Jutun ohjeet on sovellettavissa myös muihin ESP8266- ja ESP32-kehitysalustoihin.

Työvälineenä MicroPython

Mikrokontrollereita ohjelmoidaan melko yleisesti Arduino IDE -kehitysympäristöllä. Ohjelmakoodi kirjoitetaan C- tai C++-kielellä Arduino IDE:n ikkunaan, minkä jälkeen Arduino IDE kääntää koodin ja kirjoittaa sen mikrokontrollerin tai kehitysalustan flash-muistiin. Muitakin vaihtoehtoja on: eräs niistä esiteltiin Skrolli 2017.2:n jutussa *NodeMCU – avain esineverkko*, jossa ESP8266:n ohjelmointiin käytettiin NodeMCU:ta ja ajon aikana tulkittavaa LUA-ohjelmointikieltä. Tällä kertaa käytämme ohjelmointiin MicroPython-ohjelmointikieltä, jota käsiteltiin myös Skrolli 2018.1:n jutussa *Python luikertelee Pokittoon*.

Tämän jutun ohjeissa oletetaan, että tietokoneelle on asennettu Python ja Pip. Pip, eli package installer for Python, on Pythonin paketinhallinta. Jutun ohjeet on testattu Python 3:lla. Windowsilla MicroPythonin asennus ja esimerkkiohjelmistauksien testaus onnistuu myös ilman Pythonia, jos sen asennus ei syystä tai toisesta ole mahdollista.

Linux-jakeluissa Python voi olla valmiiksi asennettuna, mutta jos ei ole, se löytyy yleensä jakelun paketinhallinnasta. Debian-pohjaisilla Linuxeilla, kuten Debian, Ubuntu ja Raspbian, Pythonin ja Pipin asennus onnistuu käskyllä `sudo apt install python3-pip`. Nykyään Linux-ympäristön voi helposti asentaa myös Windows 10:een menemällä Microsoft Storeen ja etsimällä sieltä esimerkiksi Linux-jakelun nimeltään Debian tai Ubuntu ja seuraamalla kaupan ohjeita. Ominaisuutta kutsutaan WSL:ksi, eli *Windows Subsystem for Linux*. Kun jutussa jatkossa mainitaan Linux, se tarkoittaa myös WSL:ää.

Python voidaan asentaa macOS:lle esimerkiksi homebrew-paketinhallinnasta. Asenna homebrew-paketinhallinta sivun `brew.sh` ohjeiden mukaisesti. Sen jälkeen Pythonin asennus onnistuu käskyllä `brew install python`.

Windowsille Python löytyy suoraan Microsoft Storesta, mutta se on jutun kirjoitushetkellä testiversio. Sillä asennetut ohjelmat eivät muun muassa ole käytettävissä ohjelman nimellä suoraan komentoriviltä, vaan ne pitää ajaa pythonin `-m`-vivun avulla. Lataa ja asenna Python mieluummin suoraan `python.org`-sivulta. Paina asennusohjelman etusivulla täppä kohtaan "Add Python 3.7 to PATH". Windowsia käytettäessä jätä jäljempänä tulevista pip-käskyistä alun `sudo`-sana pois.

MicroPythonin asennus

Ensimmäisenä tarvitaan ESP32:lle tarkoitettu MicroPython-laiteohjelmisto. Se löytyy osoitteesta micropython.org/download#esp32. BPI-Bitissä ei ole niin sanottua SPIRAMia, joten valitse viimeisin Standard-versio.

MicroPython-firmware voidaan kirjoittaa BPI-Bitin ESP32:n flash-muistiin muutamilla eri keinoilla. Jos kirjoittaminen halutaan tehdä Windowsilla graafisella ohjelmalla, lataa ESP:n virallinen Flash Download Tools

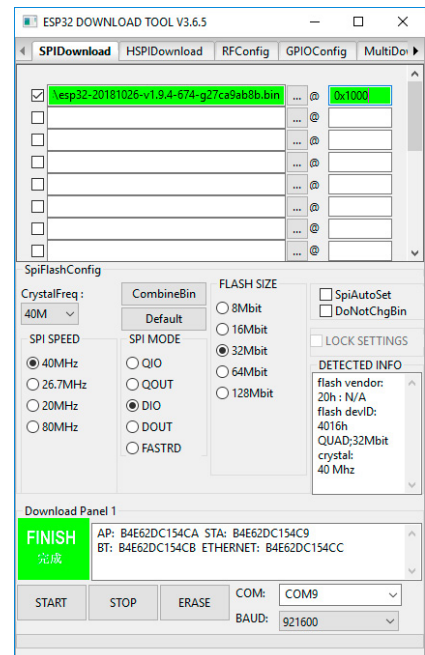
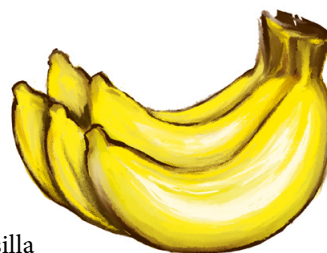
osoitteesta www.espressif.com/en/support/download/other-tools. Pura zip-tiedosto ja käynnistä `flash_download_tools_vx.x.x.exe`. Valitse käynnistyvästä ohjelmasta ESP32 DownloadTool. Valitse sitten avautuvasta ikkunasta ensimmäisenä oikeasta alareunasta COM-portti, joka on yleensä listan suurinumeroisin COM-portti. Saat hiukan vauhtia kirjoittamiseen valitsemalla nopeudeksi COM-portin alapuolelta 921600, mutta oletusnopeuskin toimii. Tyhjennä varmuuden vuoksi flash-muisti painamalla Erase. Valitse tämän jälkeen ylimmälle riville firmware-tiedosto painamalla nappia, jossa on kolme pistettä. Aseta @-merkin jälkeiseksi osoitteeksi 0x1000. Aseta firmware aktiiviseksi rivin vasemmassa reunassa olevasta valintamerkistä (kuva 1). Paina Start.

Firmware voidaan kirjoittaa myös suoraan komentoriviltä joko Windowsilla, Linuxilla tai macOS:llä `esptool.py`-nimisellä ohjelmalla. Se on Python-ohjelma, joten koneessa pitää olla Python asennettuna. `Esptool` löytyy osoitteesta github.com/espressif/esptool/releases, mutta sen voi asentaa suoraan Pythonin paketinhallinnasta käskyllä `sudo pip3 install esptool`.

Tyhjennä aluksi flash-muisti käskyllä `esptool.py erase_flash` ja kirjoita sen jälkeen firmware käskyllä `esptool.py write_flash -z 0x1000 esp32-xxxxxxx-vx.x.x.bin`. Nykyaikaisen `esptool`in pitäisi osata automaattisesti haistella sekä laitteen tyyppi että COM-portti, mutta jos se ei löydä laitetta, käytä lisäksi parametreja `--chip esp32 --port /dev/ttyUSB0`. WSL:ssä `esptool` ei löydä COM-porttia, joten anna portti parametrina esimerkiksi `--port /dev/ttyS9` (tämä on WSL:ssä Windowsin COM9).

Ensimmäinen kosketus MicroPythoniin

Yksinkertaisin ja suoraviivaisin tapa ottaa yhteys MicroPythoniin on USB-sarjaportin kautta terminaaliohjelmalla, esimerkiksi Windowsilla Puttyllä tai RealTermillä. Nopeudeksi valitaan 115200 bittia sekunnissa. Linuxilla voi käyttää esimerkiksi Minicom-ohjelmaa asentamalla sen käskyllä `sudo apt install minicom`



Kuva 1. MicroPython-firmwaren kirjoitus Windows-ohjelmalla.

ja käynnistämällä sen käskyllä `minicom -b 115200 -D /dev/ttyUSB0`. Kun yhteys on muodostettu, terminaalissa avautuu MicroPythonin sisältämä REPL-niminen komentokehote, jolle voi antaa suoraan MicroPython-komentoja. Kirjoita komentokehotteeseen esimerkiksi käsky `print("Hei Skrolli!")` ja paina enter.

MicroPython on Pythonin pikkuveli, joten molemmat kielet jakavat suuren osan syntaksista ja peruseriaatteista. Pääsääntöisesti alussa otetaan haluttu kirjasto tai moduuli käyttöön `import`-käskyllä, minkä jälkeen kyseisen kirjaston käskyt ja toiminnot ovat käytettävissä. Tämä pätee myös help-käskyyn: jos help-käskyllä yrittää pyytää ohjeistusta johonkin kirjastoon, jota ei ole `import`-komennolla ensin ladattu, help antaa virheilmoituksen. Help-komentoa kannattaa varsinkin alussa käyttää, koska sillä näkee erilaisen luokkien, funktioiden ja vakioiden nimet. Käskyllä `help()` saa yleistä tietoa, joka kannattaa lukea läpi. Käsky `help('modules')` listaa käytettävissä olevat moduulit. Käskyllä `help(kirjaston nimi)`, esimerkiksi `help(machine)`, saa listan kyseisen kirjaston luokista, funktioista ja vakioista. Kirjaston sisältämän luokan funktiot ja vakiot saa listattua käskyllä `help(kirjastonnimi.luokka)`, esimerkiksi `help(machine.ADC)`.

Toisin kuin monissa muissa kielissä, Pythonissa niin sanotuilla whitespace-merkeillä on suuri merkitys. Näitä ovat muun muassa välilyönti, sarkain ja rivinvaihto. Siinä missä esimerkiksi C:ssä tai C++:ssa erilaiset lohkot määritellään aaltosulkeilla, Pythonissa ne ilmaistaan sisennyksillä. Python-standardin mukaan sisennyksenä käytetään neljää välilyöntiä, mutta myös tabulaattori tai esimerkiksi kaksi välilyöntiä toimivat. Pythonin komentoriville käskyjä suoraan kirjoitettaessa sisennykset tulevat automaattisesti. Useita erillisiä komentoja ei pääsääntöisesti voi kirjoittaa samalle riville. Ohjelmassa ylimääräiset välilyönnit esimerkiksi tyhjällä rivillä aiheuttavat virheilmoituksen.

Merkkijonot voidaan kirjoittaa joko puolilainausmerkkien tai lainausmerkkien väliin. Ainoa ero on siinä, että jos käytät puolilainausmerkkejä, voi merkkijono sisältää lainausmerkkejä ilman, että niiden eteen tarvitsee laittaa kenoviivaa. Sama toimii myös toisin päin, eli lainausmerkkien väliin voi laittaa puolilainausmerkkejä ilman edeltävää kenoviivaa.

Koska komentoja voi kirjoittaa suoraan komentoriville, eli tässä tapauksessa MicroPythonin REPL-tulkkiin, on käskyjen kirjoitusasu ja toimivuus helppo testata yksi kerrallaan. Varsinkin alussa, kun ohjelmointi perustuu pitkälti yritys-erehdys-menetelmään, tämä säästää monelta muokkaa-tallenna-käännä-virhe-vaiheelta. REPL sisältää pienen komentohistorian, jota voi selata nuoli ylös -näppäimellä. Lisäksi REPL sisältää rivieditorin, eli käskyä voi muokata nuolinäppäimillä vasemmalle ja oikealle siirtyen. Kirjoittaessasi käskyä näet käytettävissä olevat funktiot ja muuttujat painamalla sarkainta. Sarkainnäppäin myös täydentää automaattisesti funktion tai muuttujan lopun, jos vain yksi vaihtoehto täsmää. Muuttujan arvon saa REPLissä tulostettua yksinkertaisesti kirjoittamalla muuttujan nimen ja painamalla enter.

Tehdään kokeeksi perinteinen ledin vilkutusohjelma suoraan terminaaliohjelman kautta. BPI-Bitissä pieni punainen LED on GPIO-pinnissä 18. Ohjelman voi kirjoittaa suoraan käsin rivi riviltä REPLin komentokehoteeseen. Jos ohjelma kuitenkin on tekstinä jo jossain, esimerkiksi verkkosivulla, pdf:ssä tai tekstieditorissa, pienet ohjelmat on kätevä kopioida ja liittää REPLin paste-tilaa hyväksi käyttäen. Paste-tila käynnistyy näppäinyhdistelmällä Ctrl-E. Jos luetaan Skrolliasia pdf-muodossa, voit kopioida ja liittää oheisen ohjelman paste-tilassa ja kuitata sen valmiiksi painamalla Ctrl-D. Muista, että sisennykset ovat pakollisia.

```
import time, machine
led = machine.Pin(18, machine.
Pin.OUT)
while True:
    led.value(1)
    time.sleep_ms(500)
    led.value(0)
    time.sleep_ms(500)
```

Ledin pitäisi nyt vilkkua. Kuittaamisen jälkeen ei tule komentokehote >>>, vaan ohjelman while-silmukka jää pyörimään ikuisesti. Ohjelman ajo keskeytetään painamalla Ctrl-C.

Monissa ESP32-kehitysalustoissa on erillinen reset-nappi, niin myös BPI-Bit:ssä. Reset-napin lisäksi pelkkä MicroPython voidaan resetoida REPLin komentokehoteessa näppäinyhdistelmällä Ctrl-D.

MicroPythoniin löytyy laaja dokumentaatio muun muassa osoitteesta docs.micropython.org/en/latest/reference/. ESP32- ja ESP8266-kohtaisten MicroPython-moduulien käyttöön liittyviä ohjeita esimerkkeineen löytyy osoitteista docs.micropython.org/en/latest/esp32/quickref.html ja docs.micropython.org/en/latest/esp8266/quickref.html.

MicroPythonin käyttö erillisellä hallintaohjelmalla

Kaikki tässä jutussa olevat ohjelmat voi kirjoittaa terminaaliohjelmalla suo-

raan MicroPythonin REPL-tulkkiin, kuten edellisessä kohdassa tehtiin. Pidemmän päälle tämä on kuitenkin hankalaa, ja olisi hyvä pystyä luomaan, muokkaamaan ja siirtämään tiedostoja tietokoneen ja laitteen välillä. MicroPython ei oletuksena sisällä tekstieditoria tai terminaaliohjelmien tukemia tiedonsiirtoprotokollia. Tämän takia tarvitsemme tietokoneelle ohjelman, jolla näitä tehtäviä voidaan suorittaa. Kun USB-kaapeli on kiinni tietokoneessa, MicroPythonia voidaan hallita sarjaportin kautta tai langattoman verkon yli. Tässä kohdassa käytämme sarjaporttia, ja langattoman verkon käytön käymme läpi myöhemmin.

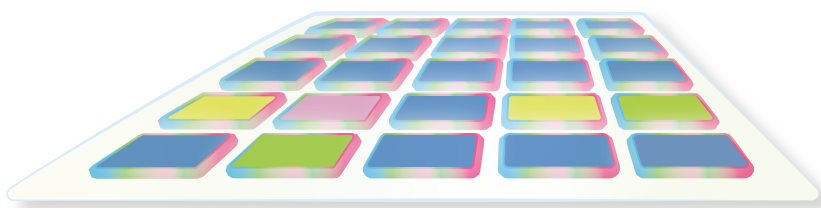
Tiedostonhallintaa sisältäviä MicroPythonille tehtyjä tietokoneohjelmia ovat muun muassa rshell, mpfshell ja ampy. Kaikki edellämainitut vaativat Pythonin toimiakseen. Asennetaan mpfshell käskyllä `sudo pip3 install mpfshell`. Mpfshellille annetaan parametrina sarjaportin nimi, esimerkiksi Windowsilla `mpfshell com9` tai Linuxilla `mpfshell ttyUSB0`.

Tärkeimpiä mpfshellin käskyjä ovat seuraavat:

- `help` - listaa käytettävissä olevat komennot
- `help käsky` - antaa käskyyn liittyviä ohjeita, esim. `help put`
- `ls` - listaa ESP32:n flash-muistissa olevan hakemiston sisällön
- `lls` - listaa paikallisen koneen hakemiston sisällön
- `cd hakemistonnimi` - vaihtaa hakemistoa ESP32:n flash-muistissa
- `lcd hakemistonnimi` - vaihtaa hakemistoa paikallisessa koneessa
- `cat tiedostonnimi` - näyttää tiedoston sisällön
- `get tiedostonnimi` - lataa tiedoston ESP32:n flash-muistista
- `put tiedostonnimi` - tallentaa paikallisessa hakemistossa olevan tiedoston ESP32:n flash-muistiin
- `repl` - REPL-tulkki (Ctrl-Q = exit)

Langaton verkko ja WebREPL

Jotta MicroPythoniin saadaan yhteys verkon kautta, pitää verkko kytkeä erikseen päälle. ESP32 voi toimia itsenäisenä langattomana tukiasema tai ottaa yhteyden olemassa olevaan langattomaan verkkoon. Olemassa olevaan langattomaan verkkoon liitytään seuraavilla käskyillä:



```
import network
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect('verkonnimi',
'verkonsalasana')
```

ESP32:n oma tukiasema saadaan päälle seuraavasti:

```
import network
ap = network.WLAN(network.AP_IF)
ap.active(True)
ap.config(essid='omatukiasema',
authmode=network.AUTH_WPA_WPA2_PSK,
password="salasana")
```

Molemmat voivat olla samaan aikaan käytössä. ESP32:n oman tukiaseman käynnistämisen viimeinen käsky ei ole pakollinen, mutta sillä voi määrittää, millä nimellä ESP32:n tukiasema näkyy, ja kytkeä WPA2-salauksen päälle. Muussa tapauksessa se näkyy muodossa "ESP_XXXXXX" ja verkko on salaamaton. Oletuksena oman tukiaseman IP-osoite on 192.168.4.1. Verkkojen ip-osoitteet näkee käskyllä `wlan.ifconfig()` ja `ap.ifconfig()`, verkkojen tilan taas käskyllä `wlan.isconnected()` ja `ap.isconnected()`. Verkot sammutetaan käskyllä `wlan.active(False)` ja `ap.active(False)`.

Seuraavaksi asennetaan WebREPL eli verkkovastakkeella (*network socket*) toimiva etäkäyttöpalvelin. WebREPL sisältyy oletuksena MicroPythonin laiteohjelmistoon, mutta se pitää ladata käskyllä `import webrepl_setup`, joka kysyy käyttäjältä muutaman kysymyksen. Yhtenä kysymyksenä on, laitetaanko WebREPL käynnistymään aina MicroPythonin käynnistyksen tai uudelleenkäynnistyksen yhteydessä. Voit vastata kyllä, jolloin asianmukaiset rivit lisätään `boot.py`-tiedostoon. Huomaa, että vielä tässä vaiheessa verkkoyhteys ei synny automaattisesti ESP32:n käynnistyksen tai uudelleenkäynnistyksen yhteydessä, koska langattoman verkon aktivointi ja asetukset eivät ole tallentuneet mihinkään.

Kun sekä verkko että WebREPL ovat päällä, laitteeseen voi ottaa yhteyden tavallisella selaimella verkkosivun micropython.org/webrepl/ kautta. Huomaa, että vaikka varsinainen verkkosivu ladataan muualta, ei liikennettä tapahdu kuin paikallisessa verkossa selaimen ja MicroPythonin välillä. Huomioida kannattaa erityisesti se, että sivusta ei voi käyttää salattua https-versiota, koska yleisimmät nykyaikaiset selaimet eivät anna

yhdistää salattua ja salaamatonta liikennettä, ja MicroPythonin kanssa liikenne tapahtuu salaamattoman verkkovastakkeen kautta. Kirjoita osoitteeksi BPI-Bitisi IP-osoite, esimerkiksi `ws://192.168.0.49:8266/`, paina Connect ja anna salasana. Tämän jälkeen voit käyttää REPLiä etänä aivan kuten terminaaliohjelmalla paikallisesti.

Myös mpfshellin pitäisi toimia verkon yli käskyllä `mpfshell ws:192.168.0.49,salasana`, mutta jutun kirjoitushetkellä en onnistunut saamaan yhteyttä.

Ohjelmien ja tiedostojen siirto ESP32:n flash-muistiin

Kahden edellisen kappaleen jälkeen meillä on kaksi vaihtoehtoista tapaa siirtää tiedostoja. Ensimmäinen vaihtoehto vaatii tiedostonhallintaohjelman (esim. mpfshell) asennuksen ja USB-kaapelin tietokoneeseen, mutta ei verkkoyhteyttä. Jälkimmäinen taas ei vaadi Pythonia mutta vaatii toimivan verkkoyhteyden, WebREPLin ja verkkoselaimen.

Asennetaan tekstieditori MicroPythoniin. Lataa tekstieditori linkistä github.com/robert-hh/Micropython-Editor/raw/master/pye.py. Siirry hakemistoon, johon latsit `pye.py`-tiedoston, ja käynnistä `mpfshell`. Tallenna `pye.py` ESP32:n flash-muistiin `mpfshell`issä käskyllä `put pye.py`.

Jos laitot verkkosetukset ja asensit WebREPLin, voit siirtää `pye.py`-tiedoston myös verkon kautta. WebREPL-sivulta löytyy Send a file -toiminto. Paina ensin Choose file, valitse `pye.py`-tiedosto ja paina Send to device.

Pye-tekstieditorin käyttö

Jotta `pye`-editorin voi käynnistää, pitää se ensin tuoda käskyllä `from pye import pye`. Tämän jälkeen se voidaan käynnistää sarjaportti-terminaalin tai WebREPLin kautta käskyllä `pye()`. Olemassa olevan tiedoston muokkaus tapahtuu käskyllä `pye("tiedostonnimi")`. Jos tekstieditoria käyttää useammin, `import`-käsky kannattaa laittaa `boot.py`:hyn, jotta se olisi suoraan käytettävissä jatkossa ilman `import`-käskyn kirjoittamista jokaisella käynnistyskerralla. Ohjeet editorin käyttöön löytyvät

Pyen Github-sivulta github.com/robert-hh/Micropython-Editor. Tärkeimmät näppäinkomennot ovat tiedoston tallennus (Ctrl-S) ja editorista poistuminen (Ctrl-Q).

FTP-palvelin

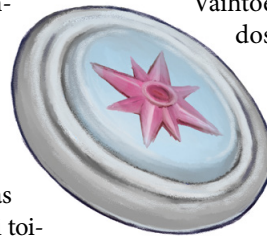
FTP on selvästi nopein tapa siirtää tiedostoja langattoman verkon välityksellä MicroPythonin ja tietokoneen välillä. Siitä on hyötyä erityisesti silloin, jos on tarve siirtää useita tiedostoja kerrallaan. Asennetaan siis FTP-palvelin MicroPythoniin. Lataa FTP-palvelinohjelma osoitteesta github.com/robert-hh/FTP-Server-for-ESP8266-and-ESP32/raw/master/ftp.py.

Siirry hakemistoon, johon latsit `ftp.py`-tiedoston, ja käynnistä `mpfshell`. Tallenna `ftp.py` ESP32:n flash-muistiin `mpfshell`in käskyllä `put ftp.py`. Vaihtoehtoisesti voit käyttää tiedostonsiirtoon WebREPLiä.

MicroPythonin FTP-palvelin on hyvin riisuttu versio, jonka merkittävin rajoite on se, että sen päällä ollessa ei MicroPythonissa voi pyöriä muita aktiivisia ohjelmia. Se ei siis ole taustalla toimiva ohjelma, joten REPL-tulkki ei ole käytettävissä FTP-palvelimen ollessa aktiivinen. Tästä syystä FTP-palvelinta ei välttämättä ole hyvä laittaa käynnistymään automaattisesti. FTP-palvelin käynnistetään käskyllä `import ftp`. `Import`-käsky toimii vain ensimmäisellä kerralla käynnistyksen jälkeen, joten jos FTP-palvelin haluat käynnistää uudestaan ilman laitteen uudelleenkäynnistystä, käytä käskyä `ftp.ftpsrvr()`. Jos käyttämäsi FTP-asiakasohjelma osaa käyttää useita samanaikaisia yhteyksiä, rajoita ne asetuksista yhteen, koska FTP-palvelin tukee vain yhtä yhteyttä kerrallaan. Tiedostojen muokkaus ja siirtely onnistuu kätevästi vaikkapa FileZilla-nimisellä monipuolisella ja ilmaisella FTP-asiakasohjelmalla.

Boot.py ja main.py

Nyt kun voimme muokata tiedostoja joko laitteeseen ladatulla tekstieditorilla tai lataamalla tiedoston ensin omalle koneelle, voimme laittaa verkkosetukset aktivoitumaan automaattisesti laitteen käynnistyessä. MicroPython ajaa aina ensimmäisenä `boot.py`-nimisen tiedoston. Tehdään siihen muuta-



ma lisäys. Lataa boot.py-tiedosto tietokoneella mpfshellillä, WebREPLillä tai FTP:llä. Jos asensit pye-tekstieditorin, voit tehdä muutokset myös suoraan ESP32:ssa käskyllä pye("boot.py"). Mpfshellillä lataus tapahtuu käskyllä get boot.py. Muokkaa tiedosto seuraavanlaisiksi:

```
import network
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
wlan.connect('verkonimi', 'verkonlasana')

def mem():
    import micropython
    micropython.mem_info()

import webrepl
webrepl.start()
```

Siirrä muokattu tiedosto takaisin ESP32:een. Nyt Wifi-yhteys luodaan automaattisesti laitteen käynnistyessä. Lisäksi luodaan mem-niminen funktio, jolla vapaan muistin määrän kysele tapahtuu helposti pelkällä käskyllä mem(). Huomaa että käskyn perään tarvitaan sulkeet. Muistakin usein käytettävistä käskyistä voi tehdä samalla tavalla oikopolun. Useimmin käytettävät moduulit ja kirjastot voi ladata automaattisesti boot.py:ssä valmiiksi import-käskyllä, mutta huomaa, että jokainen kirjasto vie muistia, jota on käytettävissä rajallinen määrä.

Boot.py:n jälkeen ajetaan aina main.py, jos sellainen on. Voimmekin nyt tehdä ensimmäisen ohjelman, joka tallennetaan ja joka käynnistyy automaattisesti. Tee seuraava main.py-niminen tiedosto MicroPythoniin asennetulla tekstieditorilla tai kirjoita ohjelma tietokoneen tekstieditorilla ja siirrä se ESP32:een.

```
import machine
led = machine.Pin(18, machine.Pin.OUT)
timer = machine.Timer(0)
def blink(timer):
    led.value(not led.value())
timer.init(period=500, mode=machine.Timer.PERIODIC, callback=blink)
```

Voit kokeilla ajaa tiedoston käsin käskyllä import main. Omat ohjelmat siis käynnistetään import-käskyllä käyttäen tiedoston nimeä ilman .py-päätettä.

Huomaa, että sisennykset ovat merkityksellisiä, eikä ohjelma toimi ilman niitä. Jos käynnistät laitteen uudelleen, pitäisi ledin vilkkua automaattisesti.

Ohjelmassa on nähtävissä, miten ESP32:n laitteistoajastinta käytetään. Ajastimia on käytettävissä muutama,

joista tässä käytetään numeroa 0. Koska ledin vilkutus tapahtuu keskeytyksellä, jää vilkkuminen päälle, vaikka ohjelmasta poistutaan. Ajastin lopetetaan käskyllä timer.deinit().

Koska import-käsky tarkistaa, onko moduuli tai ohjelma jo ladattu aikaisemmin, ei import-käsky toimi toista kertaa peräkkäin. Jotta voisit saada import-käskyn käynnistämään ohjelman uudestaan, pitää moduuli ensin poistaa muistista käskyllä:

```
import sys
del sys.modules["ohjelma"]
```

Toinen vaihtoehto moduulin poistoon muistista on resetoita MicroPython näppäinyhdistelmällä Ctrl-D. Tällöin boot.py ja main.py ladataan uudestaan. Ctrl-D on siis nopein tapa ajaa boot.py ja main.py.

BPI-Bitin ohjelmointi

Seuraavaksi käymme läpi kaikki tärkeimmät BPI-Bitin sisältämät komponentit. Jokaisen komponentin kohdalla on esimerkkiohjelma, jota voi testata sellaisenaan. Lediä on käsitelty jo aikaisemmissa esimerkeissä, joten sitä ei käydä enää erikseen läpi.

Pinnien numerot näkyvät taulukossa 1. Jostakin syystä 0:lla merkatun reunaliittimen pinni on sama kuin summerin pinni. Tämä rajoittaa 0-reunaliittimen käyttöä.

Painikkeet

Kaksi painiketta, A ja B, on kytketty ESP32:n nastoista 35 ja 27 maadoitukseen. Niitä voidaan lukea esimerkiksi käyttämällä keskeytyksiä seuraavasti:

```
import machine
def buttonA(pin):
    print("Nappi A!")
def buttonB(pin):
    print("Nappi B!")
p35 = machine.Pin(35, machine.Pin.IN, machine.Pin.PULL_UP)
p27 = machine.Pin(27, machine.Pin.IN, machine.Pin.PULL_UP)
p35.irq(trigger=machine.Pin.IRQ_FALLING, handler=buttonA)
p27.irq(trigger=machine.Pin.IRQ_FALLING, handler=buttonB)
```

Summeri

Pieni summeri on kytketty ESP32:n GPIO25-nastaan. Se päästää pienen naksauksen aina, kun jännite muuttuu, joten sitä ohjataan kytkemällä jännitettä päälle ja pois. Helpoimmin tämä tapahtuu käyttämällä pulssinleveysmodulaatiota eli PWM:ää. ESP32:n

Komponentti	GPIO
Nappi A	35
Nappi B	27
LED	18
Summeri	25
Valoanturi (vasen)	36
Valoanturi (oikea)	39
Lämpötila-anturi (NTC)	34
LED-matriisi (data)	4
LED-matriisi (enabled)	2
MPU9250/AK8963 (I2C:SDA)	21
MPU9250/AK8963 (I2C:SCL)	22
MPU9250/AK8963 (INT)	33
BPI-Bit reunaliitin (0)	25
BPI-Bit reunaliitin (1)	32
BPI-Bit reunaliitin (2)	33

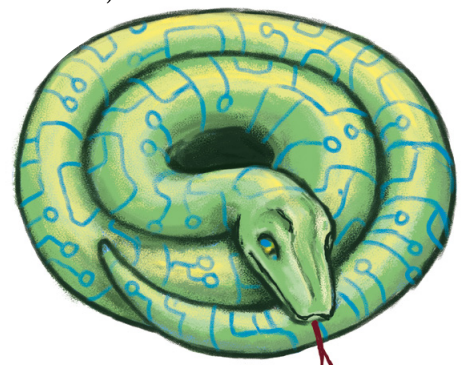
Taulukko 1. BPI-Bitin ESP32:een kytkettyjen komponenttien pinnit.

pulssisuhde (*duty cycle*) vaihtelee 0:sta (jatkuvasti pois päältä) 1023:een (aina päällä). Kun annamme duty-parametriksi arvon 512, saamme symmetristä kanttiaaltoa: puolet ajasta päällä, puolet ajasta pois päältä. Taajuus määrätään freq-käskyllä. Esimerkkiohjelma soittaa puolen sekunnin piippauksen.

```
import machine, time
pwm = machine.PWM(machine.Pin(25))
pwm.freq(500)
pwm.duty(512)
time.sleep_ms(500)
pwm.deinit()
```

Valoanturit

Valoanturit sijaitsevat led-matriisin kummallakin puolella painonappien yläpuolella. Valoanturit on toteuttu fototransistorilla ja sen kanssa sarjaan kytketyllä vastuksella, jotka yhdessä muodostavat jännitteenjakajan. Varsin puolella oleva valoanturi on kytketty ESP32:n pinniin GPIO36 ja oikealla puolella pinniin GPIO39. Valoantureita luetaan analogisisäänmenoilla eli ADC:llä (Analog-to-Digital Converter).



```
import machine, time
sensor1 = machine.ADC(machine.Pin(36))
sensor2 = machine.ADC(machine.Pin(39))
sensor1.atten(machine.ADC.ATTN_11DB)
sensor2.atten(machine.ADC.ATTN_11DB)
while True:
    print('vasen:%5d oikea:%5d' %
          (sensor1.read(), sensor2.read()))
    time.sleep_ms(500)
```

Lämpötila-anturi

Lämpöanturina toimii NTC-termistori eli vastus, jonka arvo muuttuu lämpötilan muuttuessa. Kytkenäkaaviosta käy ilmi, että termistorin kanssa sarjassa on 4K7 eli 4 700 ohmin vastus. Termistori ja vastus muodostavat jännitteenjaon ESP32:n GPIO34-pinniin. Termistorin kohdalla lukee kytkenäkaaviossa 103F3950F, jossa 103 tarkoittaa 10×10^3 eli 10 000 ohmia. Se on termistorin resistanssi 25 °C:n lämpötilassa. Lisäksi merkinnästä käy ilmi niin kutsuttu β -arvo, joka tässä tapauksessa on 3 950. Näiden tietojen lisäksi tarvitaan ESP32:n analogisisään-tulon tarkkuus, joka on 12 bittistä, eli se voi saada arvoja väliltä 0–4095. Kun analogisisään-tulon arvo on luettu, voidaan laskea termistorin resistanssi R_t , josta saadaan Steinhart–Hart-yhtälöllä varsinainen lämpötila.

```
import machine, math
adc = machine.ADC(machine.Pin(34))
adc.atten(machine.ADC.ATTN_11DB)
data = adc.read()
Rt = 4095*4700/data - 4700
temperature = round(1 / (math.log(Rt / 10000) / 3950 + (1 / (25 + 273.15)))) - 273.15
print(temperature)
```

Termistori on sikäli hiukan huonosti sijoitettu, että se lämpenee kehitysalustan komponenttien lämpenemisen takia jonkin verran. Se ei siis välttämättä edusta kovin hyvin ympäröivän tilan lämpötilaa.

Myös asentoanturi MPU9250 sisältää lämpötila-anturin. Itselläni se antoi jatkuvasti yli 5 astetta liian korkeaa lämpötilaa. Sen arvon voi lukea seuraavasti:

```
import machine, struct
i2c = machine.I2C(scl=machine.Pin(22), sda=machine.Pin(21))
data = i2c.readfrom_mem(0x69, 0x41, 2)
temperature = round(struct.unpack('>H', data)[0] / 333.87 + 21.0)
print(temperature)
```

LED-matriisi

Ledeinä on 5×5 kappaleen matriisi sisäisellä ohjauselektronikalla varustettuja WS2812B-mallin RGB-LED-elementtejä. Elementtien datapinnit ovat sarjassa, jolloin bitit siirtyvät elementiltä toiselle ja tarvitsevat koko matriisin ohjaukseen vain yhden IO-pinnin, joka on GPIO4. LED-elementtien virransyöttöä ohjaa SY6280AAC-piiri, joka muun muassa suojaa niitä ylivirtalta. Piiriin on kytketty GPIO2, jolla voidaan ohjata LED-elementit kokonaan päälle tai pois päältä.

Virransyöttö on oletuksena pois päältä, joten ensin ohjelmassa aktivoitetaan LEDien virransyöttö. Tämän jälkeen luodaan LED-nauhaobjekti. LED-nauhaobjekti on taulukko, jonka jokainen alkio muodostuu kolmesta numeroarvosta arvovälillä 0–255: punainen, vihreä ja sininen. Taulukon muokkaaminen ei vaikuta suoraan LEDien tilaan, vaan LED-nauhaobjekti saadaan päivittämään write()-komentolla. Oheinen ohjelma piirtää kuvion ja valitsee yksittäisten LEDien värit satunnaisesti kaksi kertaa sekunnissa.

Jokaisen LEDin yksittäinen väri-komponentti voi viedä 60 mW, joten kaikkien LEDien palaessa täysillä voi pelkkien LEDien virrankulutus olla $60 \text{ mW} \times 3 \times 25$ eli 4,5 W. Tämä on syytä huomioida esimerkiksi virtalähteen valinnassa, jos täyttä kirkkautta halutaan käyttää.

```
import machine, neopixel, random
np_enabled = machine.Pin(2, machine.Pin.OUT)
np_enabled.value(1)
np = neopixel.NeoPixel(machine.Pin(4), 25)
sydan = [14, 8, 2, 1, 5, 11, 15, 21, 22, 18]
while True:
    for pixel in sydan:
        r = random.randint(0,1)*50
        g = random.randint(0,1)*50
        b = random.randint(0,1)*50
        np[pixel] = (r, g, b)
    np.write()
    time.sleep_ms(100)
```

Gyroskooppi, kiihtyvyyssanturi ja kompassi

Gyroskoopin, kiihtyvyyssanturin ja kompassin virkaa hoitaa MPU9250-piiri, joka on kytketty ESP32:n nastroihin 21 ja 22 I2C-väylällä. Sen I2C-osoite on 0x69 (105). Kompassin tarvitseman AK8963-magnetometrillä I2C-osoite on 0x0c (12), ja se on sisäänrakennettu MPU9250:een. I2C-väylän käyttöönotto ja I2C-väylällä

```
# Listaa hakemiston tiedostot
def ls():
    import os
    print(os.listdir())

# Näytä tiedoston sisältö, esim.
cat("boot.py")
def cat(filename):
    file = open(filename, "r")
    while True:
        data=file.readline()
        if data=='':
            break
        print(data.rstrip())
    file.close()

# Poistaa moduulin tai ohjelman muistista, esim. unload(omaohjelma)
def unload(mod):
    mod_name = mod.__name__
    import sys
    if mod_name in sys.modules:
        del sys.modules[mod_name]

# Ajaa ohjelman, esim.
run(omaohjelma)
def run(mod):
    unload(mod)
    __import__(mod.__name__)

# Näyttää vapaan käyttömuistin määrän
def mem():
    import micropython
    micropython.mem_info()

# Näyttää vapaan flash-muistin määrän
def df():
    import os
    s = os.statvfs('/')
    return ('{0} MB'.format((s[0]*s[3])/1048576))

# Vaihtaa hakemistoa, esim.
cd("omahakemisto")
def cd(dir):
    import os
    os.chdir(dir)

# Luo hakemiston, esim.
md("omahakemisto")
def md(dir):
    import os
    os.mkdir(dir)
```

Pieniä yleishyödyllisiä funktioita boot.py:hyn lisättäväksi.

olevien laitteiden listaus onnistuu seuraavasti:

```
import machine
i2c = machine.I2C(scl=machine.Pin(22), sda=machine.Pin(21))
i2c.scan()
```

Arvon lukeminen I2C-väylällä olevasta laitteesta tapahtuu käskyllä

```
data = i2c.readfrom_mem(0x69, 0x75, 1)
```

Käskyn ensimmäinen parametri 0x69 on MPU9250:n I2C-väylän osoite. Seuraava parametri on luettavan rekisterin osoite ja viimeinen tavujen määrä.



Scan-käskey palauttaa vastauksen [105], joka on MPU9250:n osoite. Jotta myös AK8963 saadaan näkyville, pitää MPU9250:n rekisterin 0x37 arvoksi kirjoittaa 0x02. Tavun kirjoittaminen tietylle laitteelle tiettyyn rekisteriin tapahtuu käskeyllä

```
i2c.writeto_mem(0x69, 0x37, b'\x02')
```

Tämän jälkeen scan-käskey palauttaa vastaukseksi [12, 105]. Ohessa on esimerkkiohjelma, joka lukee kaikkien kolmen anturin arvot kerran sekunnissa. MPU9250 sisältää lisäksi lämpötila-anturin, jonka arvo luettiin lämpötila-anturia koskevassa kappaleessa.

```
MPU9250 = 0x69
MAG = 0x0c
import machine, ubinascii, time,
struct
i2c = machine.I2C(scl=machine.
Pin(22), sda=machine.Pin(21))
# tunnista (0x71 = MPU-9520)
dev1 = i2c.readfrom_mem(MPU9250,
0x75, 1)
print(ubinascii.hexlify(dev1))
# aseta magnetometri näkyväksi I2C-
väylälle
i2c.writeto_mem(0x69, 0x37, b'\x02')
# tunnista (0x48 = AK8963)
dev2 = i2c.readfrom_mem(MAG, 0x00, 1)
print(ubinascii.hexlify(dev2))
# aseta magnetometrin asetukset:
16-bittiset arvot ja jatkuva
lukeminen
i2c.writeto_mem(MAG, 0x0a, b'\x16')

while True:
    data = i2c.readfrom_mem(MPU9250,
0x43, 6)
    gyro = struct.unpack('>HHH',
data)
    print(gyro) # gx, gy, gz
    data = i2c.readfrom_mem(MPU9250,
0x3B, 6)
    acc = struct.unpack('>HHH', data)
    print(acc) # ax, ay, az
    data = i2c.readfrom_mem(MPU9250,
0x03, 6)
    mag = struct.unpack('>HHH', data)
    print(mag) # mx, my, mz
    time.sleep_ms(1000)
```

Struct.unpack-käskey vaativat hiukan selitystä. Kaikki kolme anturia palauttavat arvonsa kolmessa peräkkäisessä 16-bittisessä rekisterissä. Yksittäisen anturin arvot ovat siis kuusi peräkkäistä tavua, jotka unpack-käskeyllä hajotetaan kolmen alkion listaksi. Muoto ilmaistaan unpack-käskeyn ensimmäisessä parametrissa, jossa ">" tarkoittaa big-endian ja H-kirjain 16-bittistä etumerkitöntä sanaa. Käskey on tehokas tilanteissa, joissa jonnossa olevaa raakadataa pitää saada muotoiltua Pythonin ymmärtämään muotoon. Muut käskeyn tuntemat vaihtoehdot löytyvät osoitteesta docs.python.org/3/library/struct.html#struct.unpack.

MicroPythonille löytyy valmis kirjasto MPU9250:lle osoitteesta github.com/tuupola/micropython-mpu9250. Kirjaston käyttämiseksi tarvitaan kolme tiedostoa: ak8963.py, mpu6500.py ja mpu9250.py. Kirjasto käyttää I2C-osoitetta 0x68 (104), joten käyttäaksesi kirjastoa BPI-Bitin kanssa muuta mpu6500.py-tiedostossa rivillä 94 oleva osoite 0x68:sta 0x69:ksi. 🐛

LYHYET

Mailman levin demo

Keväällä se tiedettiin: Jättimäinen ultraleveän kuvasuhteen Assembly Summer 2019 -tapahtuman pääruutu olisi isointa, mitä vielä on Assemblyilla nähty. Peräti 7502 × 1500 pikselistä koostuvalle ruudulle ei kuitenkaan ollut sisältöä natiiviresoluutiolla. Taitelijakollektiivi Hackers Media Industries otti haasteen vastaan ja pyöräytti villin Real Wild -demon kompoon.

"Jättimäisten IMAXia suuremmalla pikselitiheydellä varustettujen ruutujen täyttäminen kauniilla pikseleillä on taitelijoillemme silkkaa peruskauraa", kertoo Hackers-kollektiivin edustaja **Ash Checksum** päätöksestä.

Assemblyillä demot näytetään videokaappauksina, jotta kompojen aikana olisi mahdollisimman pieni mahdollisuus mokiin. Hackers Media

Industriesin [Hacklab.fi](https://hacklab.fi)-mainos oli kuitenkin videoksi purkitettuna kohtuuttoman raskas.

"Halusimme tietysti alun perin näyttää videon mahdollisimman laadukkaana ilman pakkausta. Jostain syystä 1 Gt/s -bittivirtaisen videon toistaminen on kuitenkin vielä vuonna 2019 hieman hankalaa. Otimme siis yhteyttä pakkausspesialistiimme **Hi-Stackiin**, joka suosittelee Apple ProResin käyttämistä", Ash Checksum selittää.

Nauhoitettu 99 sekunnin demo kuitistui ProResillä 50 gigatavuun, minkä jälkeen sen näyttäminen oli helppo nakki Assemblyjen sali-av:sta vastavalle tiimille. Kun tunkkauksen jälkeen hirviökokoinen pätkä oli saatu pyörimään 30 minuuttia ennen ensiesitystä, siitä puuttui osa pikseleistä. Hacklab.

fi-mainos näytettiin tämän takia vielä varsinaisen kompon lopuksi uudelleen virheettömänä.

Demon Linux- ja Windows-versiot löytää esimerkiksi [Pouet.net](https://pouet.net)istä nimellä "[Hacklab.fi advertisement](https://hacklab.fi)", ja se toistuu myös pienemmällä ruuduilla. Kokemus tosin jäänee torsoksi. "Alkuperäisellä resoluutiolla oleva video löytyy vain Youtubesta (youtu.be/ughpeSx7u64), sillä videon koko vastaa kuulemma lähes kaikkien 2019 julkaistujen demojen kokoa yhteensä scene.orgissa", kertoo Ash.

Ash lähettää terveiset Assemblyjen sali-av:lle, compocrew'lle sekä kompoihiin osallistujille. "Toivottavasti ensi vuonna täyden resoluution tuotoksia nähdään useampikin." Skrolli yhtyy toiveeseen. 🐛

Tapio Berschewsky



Ruby

Japanin jalokivi koodaajille

Ruby on erikoinen ohjelmointikieli. Se painottaa käyttömukavuutta suorituskyvyn sijaan. Ja juuri siksi se on niin tehokas.

Teksti: Mika Haulo

Kuvat: Laura Loukola

Rubyn kehitti japanilainen Yukihiro ”Matz” Matsumoto 1990-luvun puolessavälissä. Hänen tavoitteenaan oli luoda Perliä tehokkaampi ja täysin oliopohjainen skriptikieli. Pythonista Matz ei täysin pitänyt, koska sen olio-ominaisuudet tuntuivat olevan lähinnä lisä pohjimmiltaan proseduraalisen ohjelmointikielen päälle. Ja niin syntyi Ruby, joka poimi vaikutteita Perlillä lisäksi Smalltalkista, Eiffelistä, Adasta ja Lispistä.

Ensimmäinen Ruby-versio julkaistiin japanilaisessa uutisryhmässä joulukuussa 1995. Länsimaihin se alkoi levitä muutamaa vuotta myöhemmin vuosituhannen taitteessa.

Itämaista estetiikkaa

Matsumoto ei itse myönnä liittäneensä Rubyn tarkoituksella kotimaansa kulttuurin piirteitä, mutta tarkka silmä saattaa löytää kielen konsepteista ripauksen japanilaista estetiikkaa – oli se sitten mukana tarkoituksella, alitajuisesti tai puhtaan sattuman johdosta (tämä on toki vain kirjoittajan makustelua). Esimerkiksi japanin kieliopin keskeinen käsite, teema¹, näyttäytyy myös Rubyn lohkosyntaksissa, jonka toimintaa selostetaan tarkemmin artikkelin laitekossa.

Kieliopissa teema määrää lauseiden puheenaiheen, eikä se vaihdu, ennen kuin aihepiiri vaihtuu toiseksi. Rubyssä puolestaan lohkon sisältämä koodi liittyy nimenomaan johonkin tiettyyn aiheeseen, kuten tiedoston käsittelyyn tai taulukon iterointiin. Sen sijaan teema itsessään – tai koodilohkoa kutsumuva metodi – ei määrittele tarkasti, miten aihepiirin sisällä asioita käsitellään.

Rubyn syntaksi tuntuu myös noudattelevan japanin sanajärjestystä, jossa predikaatti tulee viimeisenä. Esimerkiksi olion ilmentymän luonnissa toiminta on lopussa: Foo.new luo uuden Foo-olion ilmentymän, kun esimerkiksi Javassa vastaava rakenne olisi new Foo.

Itse asiassa oliopohjaiset ohjelmointikielien laajemminkin noudattelevat tätä sanajärjestystä. Yleisessä

¹ suomi-japani.net/?page_id=2249

syntaksissa luokan tai sen instanssin voidaan ajatella olevan ”lauseen” subjekti ja metodin nimi on lopussa oleva predikaatti: `Foo.do_magic`. Kenties Matsumoton mielenkiinto luoda *täysin* oliopohjainen kieli onkin jossain määrin perua siitä, miten hän luonnostaan käyttää äidinkieltään.

Mielekäs koodattava

Rubyn syntaksi on pelkistetyt elegantti. Se ei käytä puolipisteitä koodirivien lopussa, ja metodikutsujen sulkeita vain silloin, kun ne ovat välttämättömiä yksiselitteisyyden kannalta. Näin ollen Ruby-koodi on luonnostaan selkeän näköistä ja helppolukuista. Toisinaan Rubyä onkin kutsuttu ”toimivaksi pseudokoodiksi”. Pseudokoodiahan käytetään silloin, kun halutaan esittää ohjelmointitekniisiä ratkaisuja tuhlaimatta aikaa syntaktisten yksityiskohdientien näpertelyyn.

Visuaalinen niukkuus ei kuitenkaan tarkoita, että Ruby olisi ominaisuuksiltaan suppea kieli. Päinvastoin, Rubyssä on monia tehokkaita ominaisuuksia, joiden tarkoitus on luoda koodaajalle joustava, monipuolinen ja intuitiivinen ohjelmointiympäristö. Monien ominaisuuksien kompleksinen toteutus on vain kätetty taidokkaasti kielien käyttäjältä. Matsumoto itse on verrannut Rubyn toteutusta ihmiskehoon: päällisin puolin kovin yksinkertainen, mutta kätkee kuorensa alle paljon monimutkaisia asioita.

Matsumoton ideologia rakentaa ohjelmointikieli pääosin ihmisten tarpeita ajatellen on nykyään helppo ymmärtää. Koneteho ja muisti on halpaa. Jo kymmenen euron kuukausihintaan saa vuokrattua palvelinalustan, jonka teho riittää palvelemaan vähintään satoja käyttäjiä päivässä. Jos taas tarkastellaan esimerkiksi ammattimaisen ohjelmistotuotannon kulurakennetta, ovat henkilöstökulut useimmiten selkeästi suurin kustannus. Siksi on järkevää panostaa siihen, että ohjelmointikieli on ihmisille helppo ja tehokas käyttää. Se on arvokas työkalu siinä missä tietokone, koodieditori ja ergonominen työpistekin ovat.

Mistä alkuun?

Helpoin tapa kokeilla Rubyä on tämä selaimella toimiva työkalu: [ruby.github.io/TryRuby/](https://github.io/TryRuby/).

Ruby-tulkki toimitetaan vakiona uusien macOS-versioiden mukana.

Rubyn mielenkiintoisimmat ominaisuudet

1. KOODILOHKOT

```
["Atari", "PC", "Amigaaa!"].each do |computer|  
  puts "Paras tietokone on #{computer}"  
end
```



Koodi `do`- ja `end`-sanojen välissä on nimeltään lohko (engl. block), joka annetaan parametrina `each`-metodille. Pystyviivojen välissä olevana `computer`-sana puolestaan on lohkon saama argumentti. `Each`-metodi ei tiedä, mitä taulukolle tehdään; se keskittyy vain iteroimaan taulukon läpi. Lohkon koodi puolestaan keskittyy itse datan käsittelyyn. Koska taulukoiden iterointi on varsin yleinen rutiini, saadaan tällä erottelulla pidettyä itse kirjoitettu koodi yksinkertaisena.

Lohkolla on kaksi vaihtoehtoista syntaksia. Edellä mainitun `do...end`-tyylin lisäksi voidaan käyttää myös aaltosulkeita `{ ... }`. Vakiintunut käytäntö on, että yhdelle riville kirjoitettavissa lohkoissa käytetään aaltosulkeita ja monirivisissä `do-end`-paria.

2. JOKAINEN ASIA ON OLIO

Rubyssä joka ikinen asia on olio – mukaan lukien primitiivit kuten kokonaisluvut. Toisin sanoen numeroita voi käyttää kuten mitä tahansa muitakin olioita.

```
1.integer? # => true  
2.5.integer? # => false  
3.times { puts "foo" }
```

Huomaa myös, että kysymysmerkki on Rubyssä sallittu funktion nimissä – konventio on, että kyllä/ei-arvon palauttavan funktion nimen loppuun laitetaan kysymysmerkki. Jos taas olion metodi muokkaa oliota, loppuun laitetaan huutomerkki.

3. PALUUARVON SAA KAIKESTA

Rubyssä jokainen koodirivi ja komento palauttaa arvon. Näin ollen erillistä `return`-komentoa ei tarvitse käyttää. Metodeissa viimeinen suoritettu asia määrittää samalla sen paluuarvon.

```
def kelvin_to_centigrade(temp)  
  temp - 273.15  
end  
  
def fiilis  
  today = Time.now  
  
  if today.monday?  
    "Meh."  
  elsif today.friday?  
    "TGIF!"  
  end  
end
```



4. LUOKKIA PYSTYY VIRITTÄMÄÄN

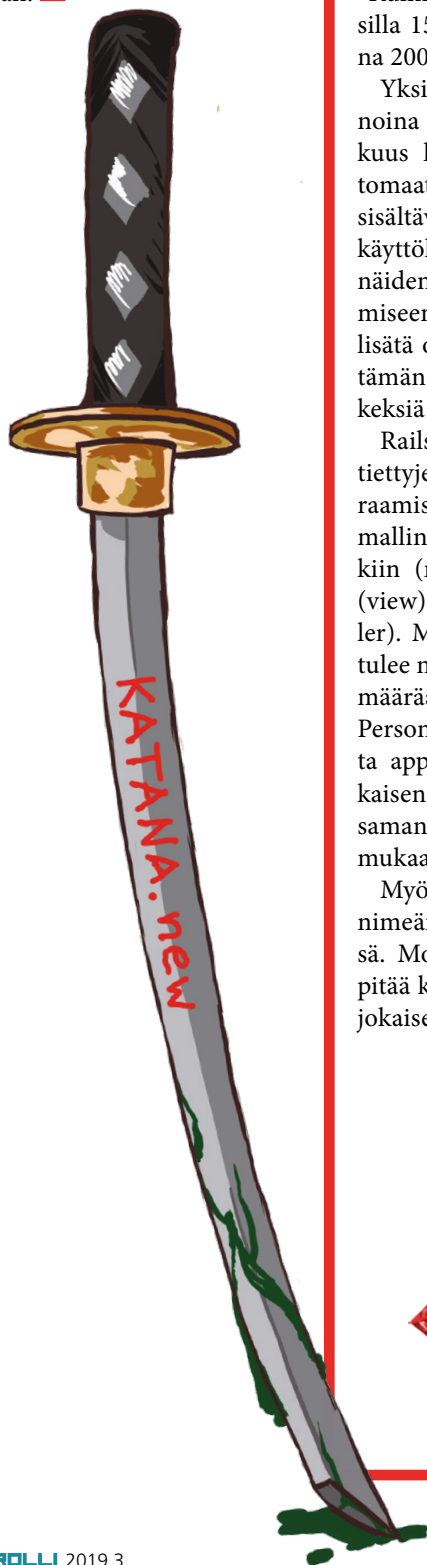
Jos Rubyn vakioluokista tuntuu puuttuvan jokin pieni mutta hyödyllisen olinen metodi, ei ole ongelma eikä mikään muokata asioita konepellin alla. Ruby sallii kielen ominaisuuksien muokkaamisen ja jatkamisen vapaasti. Alla olevassa esimerkissä `String`-luokan olioon lisätään lennossa metodi, jota standardikirjastossa ei ole.

```
lausahdus = "Miksi? No siksi!"  
  
class String  
  def teinixify!  
    self.gsub! "ks", "x"  
  end  
end  
  
puts lausahdus # => Miksi? No siksi!  
lausahdus.teinixify!  
puts lausahdus # => Mixi? No sxi!
```



Linux-jakeluissa se on vapaasti saatavilla pakettivarastoista. Windowsilla voi käyttää Ruby Installer -apuohjelmaa (rubyinstaller.org). Tarkemmat asennusohjeet löytyvät Rubyn verkkosivuilta: ruby-lang.org/en/documentation/installation/

Rubyn verkkosivujen Documentation-osiosta löytyy hyödyllisiä linkkejä Rubyn opiskelun aloittamiseen ja vinkkejä devausympäristön pystyttämiseen. Lisäksi Stack Overflow, YouTube ja monet yksittäiset blogit ovat hyviä tiedonlähteitä – ei muuta kuin googlaamaan! 📖



Jalokivi raiteilla

Ruby-ohjelmointikielen kanssa tapaa yleensä myös Ruby on Rails -webbkehitysympäristön. Se on jossain määrin yksinkertainen mutta monipuolinen ympäristö, jolla saa tehtyä mitä tahansa 15 minuutissa. Ainakin sellaisella lentävällä lauseella Railsista puhuttiin kaveripiireissä sen jälkeen, kun alkuperäinen kehittäjä julkaisi ”Kuinka rakentaa oma blogi Railsilla 15 minuutissa” -videon vuonna 2005.

Yksi syy Railsin suosion nousuun noina aikoina oli juuri sen tehokkuus koodatessa. Ympäristön automaattisesti tuottamat ”scaffoldit” sisältävät tietokannan luomisen, käyttöliittymän ja ohjauskoodin näiden yksinkertaiseen muokkaukseen. Ohjelmoijan tarvitsee vain lisätä omat hienoudet ja muotoilut tämän päälle, eikä pyörää tarvitse keksiä tyhjistä asti.

Railsin tehokas käyttö edellyttää tiettyjen käytäntöjen tarkkaa seuraamista. Koodi jakautuu MVC-mallin mukaisesti tietokantaluokkiin (model), käyttäjän näkymiin (view) ja ohjauskoodiin (controller). MVC-mallin eri osien koodit tulee nimetä ja sijoittaa konvention määräämällä tavalla. Esimerkiksi Person-luokan malli löytyy polusta `app/models/Person.rb`. Kun jokaisen Rails-projektin rakenne on samanlainen, on helpompaa hypätä mukaan uuteen projektiin.

Myöskin luokkien ja metodien nimeämiselle on omat käytäntönsä. Monisanaisten luokkien nimet pitää kirjoittaa CamelCasella, jossa jokaisen sanan ensimmäinen kir-

jain on isolla. Metodien nimet kirjoitetaan aina snake_casella, eli alaviivoin erotetuilla sanoilla. Railsin automatiikka perustuu siihen, että nimeämiskäytäntöjä noudatetaan. Näistä käytännöistä on mahdollista poiketa, mutta silloin se pitää erikseen kertoa koodissa.

Pysyvää talletusta kaipaavat oliot tallennetaan tietokantaan. Rails osaa luoda taulun automaattisesti, ja taulun nimeksi tulee sitä vastaavan luokan nimen monikko. Esimerkiksi jos tehdään luokka *Customer*, niin Rails tallentaa sen oliot taulukkoon, jonka nimi on *customers*. Rails osaa myös englannin kielen epäsäännölliset taivutusmuodot: esimerkiksi luokan *Person* taulukon nimeksi tulee *people*. Jos automaattinen taivutus menee pieleen, ohjelmoija voi antaa oikean monikkomuodon käsin.

Jossain vaiheessa Rails oli kenties suosituin webbikehtisympäristö, mutta viime vuosina muut ovat ajaneet siitä jo ohitse. Stack Overflow'n kyselyssä¹ keulassa ovat Javascript-pohjaiset ympäristöt, mutta Rails keikkaa vielä listoilla mukana. Viimeisin Railsin iso päivitys (versio 6.0) julkaistiin hetki sitten, elokuussa 2019, joten ympäristönä se on vielä elinvoimainen ja kehityksessä mukana. Rubyn lisäpalikoiden eli ”gemien” kehitystä tapahtuu myöskin, joskin iso osa kehittäjistä on hypännyt muiden kielten vietäväksi ja osa gemeistä on jäänyt sammaloitumaan.

**Anssi Kolehmainen,
Jarno Niklas Alanko**



¹ insights.stackoverflow.com/survey/2019#technology

AVARUUSMARSU hurmasi lapset

Lapsilta saa ennakkoluulottomia ideoita ja suoraa palautetta myös koodausprojektiin.

Ronja Koistinen



Olen pitkään halunnut opetella pelikoodaamista, mutta hyvä idea on uupunut. Kesälomalla sain viimein innostavan idean: teen sivulta kuvatun shoot-em-upin 1980-luvun japanilaiseen tyyliin. Sellaisen kuin lapsena enon MSX:llä pelaamani *Nemesis* (tunnetaan myös nimellä *Gradius*).

Parissa päivässä peli eteni hyvään vaiheeseen: oli jo sivuttain vierivä tähtikenttä, satunnaisesti syntyviä vihollisia, avaruusalus ja nappi, jota painamalla alus ampui. Törmäyksen tunnistusta oli sen verran, että viholliset kuolivat ammuksiin, mutta ne eivät ampuneet takaisin, eikä pelissä ollut mitään tapaa pelaajan kuolla.

Eräänä iltana olin tuttavaperheen luona vierailulla ja keksin näyttää keskeneräistä peliäni viisi- ja yhdeksänvuotiaalle lapsille. Vaikkei pelissä vielä voinut tehdä muuta kuin lennellä ympäriinsä ja ampua satunnaisesti ruudun oikeasta laidasta syntyviä passiivisia vihollisia, lapsille riitti moneksi tunniksi riemua pelistä. Piti ihan vahvia vuorottelua.

Näytin myös vähän, miten peli toimii. Piirsimme aluksi uudet kuvat peliin. Korvasimme viholliset karkeasti pikapiirretyillä *Tähtien sodan* TIE-hävittäjillä. Sitten lisäsimme vihollisten

määrää. Alkuperäisessä koodissa oli 40 prosentin todennäköisyys synnyttää uusi vihollinen kerran sekunnissa – muutimme koodia niin, että joka framella synnytettiin yksi uusi vihollinen. Lapsia nauratti absurdi TIE-hävittäjien hyttysparvi, joka alkoi pursua pelikentän laidalta.

Lopuksi vielä korvasimme pelaajan avaruusaluksen marsulla. Perheessä on nimittäin marsuja lemmikkeinä, ja niinpä tutun karvabataatin siirtäminen pelin sankariksi resonoi lapsissa hulluvastaisesti. Näytin, kuinka marsuksi saadaan valokuva Googlen kuvahausta poimimalla, GIMPillä oikeaan kokoon skaalaamalla ja kuvan taustaa läpinäkyväksi pyyhkimällä.

Silmiä avaava kohtaaminen

Kokemus oli kaikin puolin antoisa. Lapsille oli varmasti kehittävää nähdä, että tietokonepelejä voi tehdä heille ennestään tuttu ihan tavallinen ihminen. Tuon ikäisille ei toki mikään koodi juuri aukene, mutta spritejen piirteleminen ja niiden ilmaantuminen pelin sisälle varmasti käyvät järkeen.

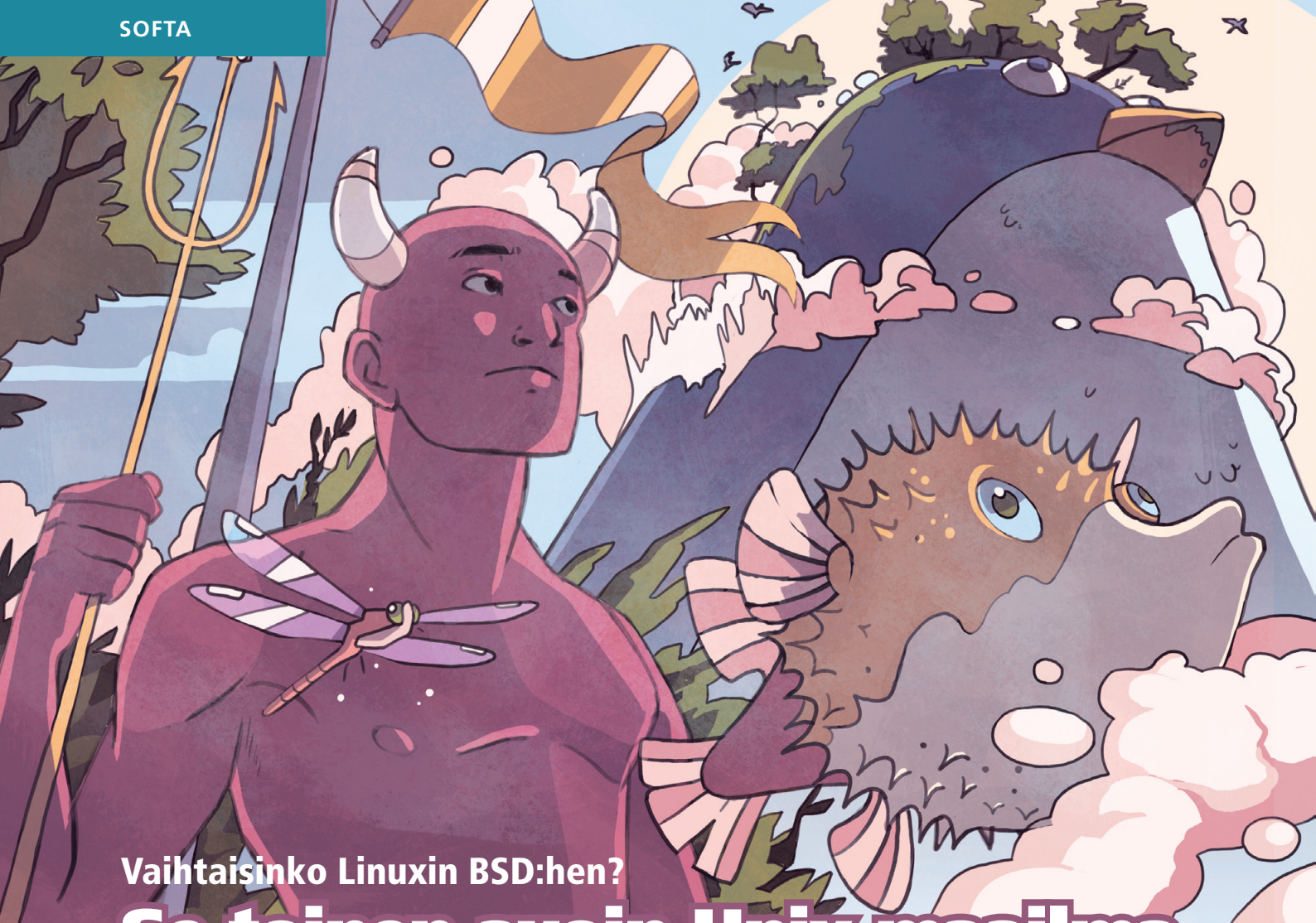
Minulle taas oli tärkeää kokemus siitä, että itse kirjoittamastani koodista oli aidosti iloa jollekulle toiselle. Lisäksi lasten kanssa pelin tarkasteleminen

sai luovuuteni lentoon ihan uudella tavalla. Idea marsusta oli omani, mutta ei se varmasti olisi pälkähtänyt päähäni, jos olisin jäänyt pelini kanssa yksin kotiin ikuisiksi ajoiksi. Olin toki jakanut linkkiä aikuisille kavereille verkkossa, mutta lapsilta saa aivan eri tavalla sievitelemätöntä palautetta ja ennakkoluulottomia ideoita.

Luulen, että marsu jää pelin sankariksi. Tähtienvälinen kybermarsu on ainakin vähän omintakeinen shoot-em-up-pelin päähahmo. Ehkä siitä riittää sen verran charmia, että joku haluaa pelata peliäni jokusen tovin. Toki piirrän marsun spriten uudestaan – ei ole hyvän tavan mukaista julkaisuta tökerästi muokattuja epämääräisiä kuvahakuosumia. TIE-hävittäjätkin vaihtuvat johonkin kybermarsujen luonnollisiin vihollisiin, vaikkapa kyberkissoihin tai -haukkoihin.

Niin muuten, peli on koodattu Rust-kielillä (rust-lang.org), josta tykkään kovasti. Projektin tarkoituksena oli listiä kaksi karpästä yhdellä iskulla: Rust ja peliohjelmointi. Tähän mennessä tavoitteessa on pysytty mallikkaasti.

Pelin nimi on (ainakin toistaiseksi) *Sideways*. Sen lähdekoodin voi ladata osoitteesta github.com/ronjakoi/sideways. 🚀



Vaihtaisinko Linuxin BSD:hen?

Se toinen avoin Unix-maailma

Jos Linus Torvalds ei olisi juuri oikeaan aikaan kirjoitellut iltapuhteeksi omaa 386:lla pyörivää Unix-kerneliään, suosittu avoimen lähdekoodin käyttöjärjestelmät saattaisivat nykyään näyttää vähän toiselta.

Teksti: Ronja Koistinen
Kuva: Emma Kantanen

Alkuperäinen *Berkeley Software Distribution* oli Unix-käyttöjärjestelmä¹, jota kehitettiin ja julkaistiin Kalifornian yliopistolta, Berkeleyyn kampukselta käsin. Järjestelmän ensimmäinen versio julkaistiin vuonna 1977 ja viimeinen vuonna 1995. BSD perustui suoraan alkuperäiseen, AT&T:ltä lisensoituun Unixin lähdekoodiin.

¹ *Unix on tavaramerkki, joka viittaa rajattuun joukkoon käyttöjärjestelmiä. Tavaramerkin ulkopuolisista käyttöjärjestelmistä käytetään monesti nimitystä Un*x- tai *nix-käyttöjärjestelmät tai Unixin kaltaiset käyttöjärjestelmät. Luettavuuden nimissä viittaamme tässä artikkelissa Unix-käyttöjärjestelmällä sekä tavaramerkin alaisiin että sen ulkopuolisiin käyttöjärjestelmiin. BSD:llä ei ole Unix-tavaramerkkiä, vaikka se onkin esimerkiksi Linuxia lähempänä alkuperäistä Unixia.*

Varsinaisen BSD:n varhishistoria ennen 1990-lukua on sinänsä mielenkiintoista, mutta tässä artikkelissa ei paneuduta siihen. Historian voi lukea haluamassaan syvyydessä internetistä aloittaen vaikkapa Wikipediasta. Sen sijaan tässä artikkelissa on tarkoitus pohtia myöhempien BSD-johdannaisien avoimen lähdekoodin käyttöjärjestelmien asemaa suhteessa 1990-luvun alussa syntyneeseen Linuxiin. Jutussa käydään läpi BSD- ja Linux-käyttöjärjestelmien eroja keskenään muun muassa kehitysprosessien, yhteisöjen, teknisten rakenteiden, filosofioiden ja lisenssikäytäntöjen suhteen.

Artikkelista saattaa paistaa läpi se, että kirjoittajalla on eniten henkilökohtaista kokemusta *OpenBSD*:stä.

Kolme suurinta jälkeläistä

Koska alkuperäinen BSD sisälsi AT&T:ltä lisensoitua epävapaita koodia, Berkeleyn Computer Systems Re-

search Groupissa oli painetta kehittää järjestelmästä versio, joka olisi täysin vapaa tästä AT&T:n koodista. Vuosina 1989–1991 vietiin läpi projekti, jossa kirjoitettiin uudestaan melkein kaikki nämä epävapait osat lähdekoodista, ja voitiin julkaista BSD-versiot *Networking Release 1* eli *Net-1* sekä *Net-2*.

Pian *Net-2*:n julkaisun jälkeen muuan aviopari **William** ja **Lynne Jolitz** päätti uudelleenkirjoittaa loputkin BSD:n epävapaita AT&T-peräisestä koodista. He myös siirsivät käyttöjärjestelmän Intel 80386 -arkkitehtuurille ja julkaisivat oman järjestelmänsä nimellä *386BSD*.



FreeBSD®

368BSD:n kehitys kuitenkin eteni hitaasti, ja jo syksyllä 1993 joukko järjestelmän käyttäjiä perusti oman haaraprojektinsa nimeltä *FreeBSD*. Se on tänä päivänä laajimmin käytetty BSD-käyttöjärjestelmä.



Suurin piirtein samoihin aikoihin perustettiin myös *NetBSD*-projekti, samaten perustuen 386BSD:hen ja osittain myös suoraan Net-2:n koodiin. NetBSD:n pääperiaatteisiin kuului alusta alkaen hajautetusti internetissä tapahtuva kehittämistyö sekä yleisölle avoin lähdekoodisäilö (repository). Jälkimmäinen ei ollut vuonna 1993 ollenkaan itsestäänselvyys edes vapaan koodin ohjelmistoprojekteissa.



Myöhemmin eräs NetBSD:n perustajajäsen, **Theo de Raadt**, erotettiin projektista erimielisyyksien vuoksi. Vuonna 1995 hän perusti oman projektinsa OpenBSD:n uutena haarana NetBSD:stä.

Painopisteet

Kolme suurinta eli Free-, Net- ja OpenBSD ovat vuosien varrella profiloituneet hieman erilaisiin painotuksiin päämäärissään.

NetBSD tunnetaan siitä, että se tukee erittäin laajaa kirjoja erilaisia tietokoneita. Sitä voi ajaa tavallisten IBM-yhteensopivien PC-laitteiden lisäksi muun muassa eri-ikäisillä Amigoilla ja Macintoshilla, Sega Dreamcastilla, Itaniumilla, VAXilla, Sparcilla sekä Alphalla. NetBSD:n käyttöä vuosituhannen alun Apple

iBookissa käsiteltiin Skrollin numerossa 2019.2.

OpenBSD:n keskipisteenä on tinkimättömyys tietoturvasa. Se ei esimerkiksi tätä nykyä käytä oletuksena ollenkaan Hyper-threadingia – varoimenpiteenä joitain Meltdown- ja Spectre-tyyppisiä spekulatiivista koodinsuoritusta hyödyntäviä haavoittuvuuksia vastaan. OpenBSD myös innonoi jatkuvasti tietoturvan saralla: viime vuosina järjestelmä on esimerkiksi alkanut linkata järjestelmäytimen kuvan (*kernel image*) uusiksi jokaisessa bootissa järjestäen kaikki symbolit satunnaiseen järjestykseen. Näin on erittäin epätodennäköistä, että missään ajossa olevassa OpenBSD-järjestelmässä olisi ytimen muistialue samassa järjestyksessä kuin jossain toisessa. Tästä syystä erilaisilla muistivuotohyökkäyksillä on vaikeampi saada aikaa mitään hyödyllisiä murtoja. Muihin käyttöjärjestelmiin sittemmin levinnyt ajettavien ja kirjoitettavien muistisivujen erottelun pakottava W^X-teknologia on sekin peräisin OpenBSD:stä.

OpenBSD-projektin ja -säätiön suosijissa kehitetään myös eräitä työkaluja ja kirjastoja, jotka ovat käytössä muuallakin: esimerkiksi *OpenSSH*, *OpenBGPD* ja *LibreSSL*.

FreeBSD on tilastojen mukaan nykyaikaisista BSD-järjestelmistä käytetyin. Se ei ole erityisen profiloitunut mihinkään erityiskäyttöön, vaan pyrkii olemaan hyvä yleiskäyttöjärjestelmä. Sitä käytetään palvelin- ja sulautetuissa ympäristöissä, pöytäkoneissa ja läppäreissä. Erityisen suosittu FreeBSD (ja siihen perustuva FreeNAS) on tiedostopalvelimissa nykyaikaisen ZFS-tiedostojärjestelmän ansiosta. ZFS on edistyneempi tapa toteuttaa tehokkaita ja vikasietoisia tallennusjärjestelmiä kuin esimerkiksi perinteinen RAID, ja sen tuki ja työkalut ovat omasta mielestäni erityisen kehittyneet FreeBSD:ssä.

Muutamia mainitsemisen arvoisia FreeBSD-johdannaisia käyttöjärjestelmiä ovat *TrueOS* (aiemmin *PC-BSD*), *FreeSBIE* (live-CD) ja *pfSense* (palo-muuri- ja reititinohjelmisto).



DragonFlyBSD

Kolmen suurimman ja niiden johdannaisten lisäksi pitää mainita vielä *Dragonfly BSD*, joka on perustettu vuonna 2004. Sen on tarkoitus olla looginen jatkumo alkuperäiselle 4.x-sarjan BSD:lle. Dragonfly pyrkii olemaan tehokas, nykyaikainen käyttöjärjestelmä, jossa on alusta asti erityisesti sille suunniteltu HAMMER-tiedostojärjestelmä ja erityisen hyvä moniprosessointituki. Dragonfly poikkeaa kolmesta suuresta siinä, että se tukee vain x86-64-arkkitehtuuria.

Lisenssi

Yksi olennainen ero Linuxin ja BSD-järjestelmien välillä on lisenssi. Linux on julkaistu GNU General Public License- eli GPL-lisenssillä (versio 2), kun taas BSD-järjestelmät käyttävät BSD-lisenssiä.

GPL on niin sanottu *copyleft*-lisenssi, eli se takaa sekä sen alaisena julkaisulle teokselle että siitä johdannaisille teoksille samat ehdot. Keskeisenä ehtona on, että teoksen käyttäjän on voitava muokata ja parantaa sitä. Tästä seuraa, että esimerkiksi Linux-ydintä autojen tai digiboksien osana myyvän yrityksen on vähintään nimenomaisesti

pyynnöstä luovutettava käyttäjän saataville tuotteessa toimitettua Linux-ydintä vastaava lähdekoodi.

GPL-lisenssiä kutsutaan näiden ominaisuuksiensa vuoksi myös ”tarttuvaksi” (contagious) lisenssiksi, sillä kaikki jatkokehitetyt

ja johdannaiset teokset on julkaistava samalla lisenssillä.

BSD-lisenssi sen sijaan sallii julkaista johdannaiset myös eri lisenssillä, eikä se velvoita johdannaisteosten lähdekoodin luovuttamiseen. Siksi BSD-järjestelmiä ja niiden koodia on käytetty ja käytetään paljon kaupalli-

"FreeBSD on tilastojen mukaan nykyaikaisista BSD-järjestelmistä käytetyin."

sisä, suljetun lähdekoodin tuotteissa. Esimerkiksi macOS perustuu paljolti BSD-järjestelmien koodille, ja Microsoft Windowsin TCP/IP-pino on Windows 2000:sta alkaen sisältänyt paljon BSD-peräistä koodia. Jatkokehitetty FreeBSD on käyttöjärjestelmänä myös Sonyn pelikonsoleissa PlayStation 3 ja 4.

Free Software Foundation on kuitenkin määritellyt sekä kaksi- että kolmilausekkeisen BSD-lisenssin avoimen lähdekoodin lisenssiksi ja yhteensopivaksi GPL:n kanssa.

Linux on vain ydin

Kuten **Richard Stallman** väsymättömästi meille muistuttaa, Linux on vain yksi osa niitä käyttöjärjestelmiä, joiden sisällä se sykkii. Monille on tärkeää kutsua näitä järjestelmiä *GNU/Linuxiksi*, sillä Linux on vain ydin (kernel), ja käyttäjätilan (user space) perustana ovat alusta asti olleet GNU-projektin työkalut kuten Bash-komen-

totulkki ja muut palikat, jotka kaikista Unix-järjestelmistä löytyvät.

Nykyajan GNU/Linux-järjestelmässä on paljon myös muita osasia, joita kehittävät sekalaiset erilliset tiimit: esimerkiksi *Systemd*, *D-Bus*, *glibc*, *X Window System* ja *Wayland*.

BSD:t ovat sen sijaan paljon tiiviimmin yhden projektin ylläpitämiä, kehittämiä ja julkaisemia kokonaisia käyttöjärjestelmiä. Esimerkiksi FreeBSD:ssä ydin, komentorivitulkki ja -työkalut, palomuurityökalut, tiedostojärjestelmäajurit ja virtualisointityökalut ovat kaikki kiinteä osa käyttöjärjestelmää.

Jos GNU/Linuxissa haluaa pystyttää kevytvirtualisoidun kontin (*container*) jollekin ohjelmistolle, mielekkäin menettely on käyttää erillisen kaupallisen yrityksen julkaisemaa *Docker*-työkalua. FreeBSD taas on jo 20 vuotta tarjonnut sisäänrakenn-

nettuna *jails*-nimisen alijärjestelmän samankaltaisten ympäristöjen hyödyntämiseen. Sen käyttöohjeet ovat oma lukunsa käyttöjärjestelmän käsikirjassa (*FreeBSD Handbook*). Jails-ympäristöt on myös integroitu suoraan esimerkiksi ZFS-tiedostojärjestelmää koskeviin työkaluihin. Myöskään jails-työkalujen ja FreeBSD-ytimen välisistä versioristiriidoista ei koskaan tule päänvaivaa, toisin kuin Linuxin ja Dockerin kanssa saattaa tapahtua. Ytimen ja jails-alijärjestelmän sisältyminen kiinteästi samaan koherenttiin ohjelmistoprojektiin takaa sen, että tiettyssä julkaisuversiossa molemmat ovat varmasti yhteensopivat.

Lähdekoodi paketinhallintana

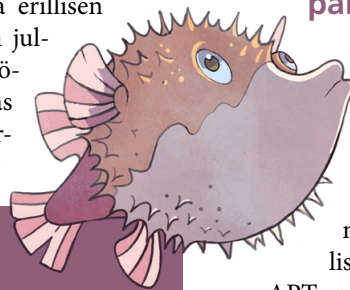
GNU/Linux-käyttäjälle paketinhallintajärjestelmät ovat tuttua kauraa. Esimerkiksi Debian-johdannaisissa jakeluissa on APT-niminen paketinhallinta, josta lisäohjelmistot asennetaan.

APT pitää myös huolen asennettujen ohjelmien ajantasaisuudesta tietoturvan ja vakauden suhteen. Komento `apt-get upgrade` päivittää kaikki järjestelmän osaset näppärästi kerralla.

BSD:ssä käyttöjärjestelmän päivitykset ja lisäohjelmistot on perinteisesti asennettu lähdekoodista. Karkeasti ottaen:

1. Käyttäjä hakee CVS-versionhallinnasta käyttöjärjestelmän lähdekoodin polkuun `/usr/src`.
2. `cvsup` -tyyppisellä komennolla haetaan ajoittain uusin versio lähdekoodista.
3. `make` ja `make install` kääntävät muuttuneen lähdekoodin ja asentavat uudet ohjelmat aloilleen.
4. `reboot` käynnistää koneen uudelleen ottaen käyttöön mahdollisesti päivittyneen ytimen.

Myös lisäohjelmat ovat perinteisesti toisessa, yllä olevan kaltaisessa CVS-säilössä. Sen nimi on NetBSD:ssä `pkgsrc`, Open- ja FreeBSD:ssä `ports`. Esimerkiksi OpenBSD:ssä, kun `ports`-metatietopuu on asennettu oikeaan paikkaan, VLC-videosoitin asennus onnistuu komennolla `make install` hakemistossa `/usr/ports/x11/vlc`. Komento



BSD-lisenssit

Yleisin nykyään käytössä oleva BSD-lisenssi on niin sanottu kolmen lausekkeen lisenssi. Sen koko teksti on seuraava:

Copyright (c) <year>, <copyright holder>

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name of the <organization> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL <COPYRIGHT HOLDER> BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Jotkut projektit, esimerkiksi FreeBSD, käyttävät riisutumpaa kahden lausekkeen lisenssiä. Siitä puuttuu kolmas lauseke, jossa kielletään käyttämästä tekijänoikeuden haltijan tai ohjelmiston kehityksessä avustaneiden (*contributors*) nimiä johdannaisteosten mainonnassa. Joitain muitakin pieniä eroja on eri lisenssien välillä, ja niihin on tarvittaessa tutustuttava tapauskohtaisesti.

kääntää ohjelmiston lähdekoodista ja asentaa paikoilleen sen binäärit, asetus-tiedostot, manuaalisivut ynnä muut.

Jos tällainen lähdekoodin kanssa puljaaminen kuulostaa oudon tutulta, syynä on ehkä Gentoo. Se on Linux-jakeluversio, jossa kaikki ohjelmistot asennetaan portage-nimisen paketinhallinnan avulla lähdekoodista kääntämällä. Käytäntö on lainattu nimenomaan BSD:stä. Tapa levittää ohjelmia lähdekoodina juontaa juurensa aikaan, jolloin Unix-koneiden suorittimien joukko oli nykyistä kirjavampi, eikä binääriyhteensopivuus ollut mitenkään taattua.

Nykyisin tosin BSD:tkin jakelevat sekä kolmannen osapuolen ohjelmistoja että itse käyttöjärjestelmän tietoturvapäivityksiä valmiiksi käännettävinä binääreinä. Työkalujen nimet ja tarkat ohjeet vaihtelevat BSD:stä toiseen. OpenBSD:n dokumentaatio, jota tutkailin kertaukseksi tätä kirjoittaessani, suosittelee nykyään käyttämään esikäännettyjä binäärejä ja sanoo, että ports on tarkoitettu edistyneille käyttäjille.

Marginaalissa jää aina jostain paitsi

Nykyajan Linux-käyttäjä on kenties jo tottunut siihen, että 3D-kiihdytys toimii heittämällä, Steamista on natiivi asiakasohjelma saatavilla ja Netflixiä voi katsoa Chromella. Linuxille on tätä nykyä saatavilla paljon kaupallista, suljetun lähdekoodin softaa – sekä pelejä että hyötyohjelmia. Winen ja Protonin kautta toimii suuri määrä Windows-softastakin.

BSD-järjestelmillä on näiden suhteen paljon hiljaisempaa. Netflixin, Spotifyn ja monen muun suoratoistopalvelun käyttämä Widevine-kopiosuojauskirjasto on saatavilla valmiiksi käännettynä binäärinä vain Windowsille, macOS:lle ja Linuxille². Jos siis päättää asentaa koneeseensa BSD:n, Netflix jää katsomatta, Spotify kuuntelematta ja suurin osa peleistä pelaamatta. Sinänsä hieman ironista, että tietävästi Netflix käyttää palvelininfarrassaan laajasti FreeBSD:tä.

Ainakin FreeBSD:ssä on tosin jonkin verran olemassa yhteensopivuus-toiminnallisuutta Linux-binääreille, mutta siihen ei kannata luottaa liiaksi.

² Widevine on tietysti saatavilla myös Androidille, iOS:lle ja niille sulautetuille järjestelmille, jotka on tungettu nykyajan älytelkkareihin.

Periaatteessa ehkä olisi mahdollista tehdä Widevinellekin jonkinlainen sovitinpalikka, jotta Chromiumin saisi lataamaan sen FreeBSD:ssä, mutta en pidätä hengitystäni.

ZFS voi mullistaa levypalvelimet

Alun perin Sun Microsystemsilla kehitetty ZFS on uuden sukupolven tiedostojärjestelmä, joka on mielenkiintoinen teknologia. Sen tarkoituksena on yhdistää tiedostojärjestelmän, fyysisten massamuistien ja loogisten osioiden hallinnoinnin käsitteet yhteen ja samaan kokonaisuuteen. Toisin sanoen ZFS:llä voi korvata kokonaan sellaiset asiat, jotka Linux-käyttäjälle ovat tuttuja termeissä RAID, LVM ja Ext4.

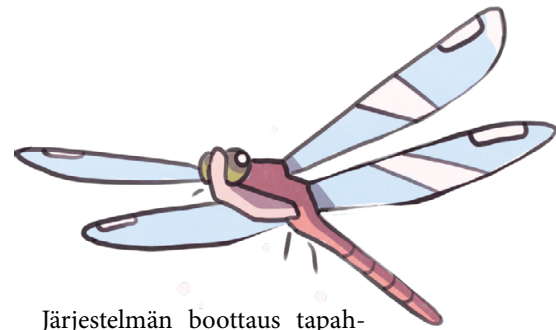
En käy tässä kovin laajasti läpi ZFS:n ominaisuuksia, mutta niihin kuuluu esimerkiksi varmuuskopiointi snapshotien avulla, copy-on-write, automaattinen datan eheyden tarkastus, RAID-tyyppinen redundanssi sekä datan deduplikaatio.

"Monella tavalla BSD:t ovat jonkin verran vanhanaikaisempia kuin GNU/Linux-jakelut."

ZFS on (Sunin ja Solariksen hiipumisen jälkeen) omasta mielestäni erityisen hyvissä kantimissa FreeBSD:ssä. Mainitsin jo edellä siitä, kuinka hyvin se on integroitu muihin käyttöjärjestelmän työkaluihin, esimerkiksi NFS:ään (verkkolevyjakoihin) sekä jail-kontteihin. Linuxiin ja macOS:ään ZFS ovat vasta tuloillaan, kun taas BSD-maailmassa se on täyttä nykyaikaa.

Perinteet kunniaan

Monella tavalla BSD:t ovat jonkin verran vanhanaikaisempia kuin GNU/Linux-jakelut. Tämän näkee yleensä heti asennusohjelmasta. Graafiset asennuskäyttöliittymät ovat harvinaisia, ja usein pääsee heti alkajaisiksi käsin osioimaan levyä fdiskillä ja alustamaan osioita.



Järjestelmän boottaus tapahtuu BSD-järjestelmissä perinteisillä shell-skripteillä, ja yleiset konfiguraatiot asetetaan tavallisesti /etc/rc.conf-tyyppisessä tekstitiedostossa. GNU/Linuxissa taas on nykyään yleensä Systemd-niminen alijärjestelmä, joka on arkkitehtuuriltaan merkittävästi erilainen ja tekee verrattain paljon asioita järjestelmässä.

Työkalut ja komentoriviympäristöt ovat tavallisesti myös GNU/Linuxiin verrattuna konservatiivisia. Esimerkiksi oletuskomentotulkki OpenBSD:ssä on rootille sh ja tavalliselle käyttäjälle ksh – FreeBSD:ssä sh ja csh. Bashin saa asentamalla paketinhallinnasta erikseen.

Monet tutut työkalut eivät tue kaikkia samoja vipuja kuin GNU-versiot tai toimivat muuten eri tavalla. Find-työkalun BSD-versio esimerkiksi haluaa aina eksplisiittisen hakupolun, kun taas GNU find hakee työhakemistosta, jos hakupolku puuttuu komennosta. Tällaisia pieniä eroja on paljon, ja ne aiheuttavat jatkuvasti pieniä epäyhteensopivuuksia shell-skripteissä, jos niiden kirjoittaja on olettanut GNU-versiot työkaluista.

Onneksi on olemassa POSIX-standardi, jota BSD-työkalutkin noudattavat. Jos tietää, mitä asioita POSIX-järjestelmälle takaa, tietää yleensä myös, mitä sekä GNU/Linuxille että BSD:lle voi olettaa yhteiseksi.

Moni on sitä mieltä, että BSD:t ovat aidommin ja puhtaammin Unixeja kuin GNU/Linux. Väitettä tukee esimerkiksi suora historiallinen yhteys AT&T:n Unixiin. Järjestelmäarkkitehtuurin ja perustyökalujen konservatiivisuus ovat toinen perustelu. Jos Unix-oikeamielisyys on sielua lämmittävä asia, BSD:hen kyllä kannattaa tutustua. Ja vaikkei olisikaan, kokemus on antoisa näkymänä erilaiseen skeneeseen ja erityyppisiin tapoihin ylläpitää, kehittää ja jaella avoimen lähdekoodin softaa.

Vähintäänkin kurkistus BSD:hen panee terveellisesti kyseenalaistamaan ja katsomaan kriittisesti GNU/Linux-jakeluiden valintoja ja suuntia. 🦋

Jonnet ei muista

Kun yhteiskunnan muutosvauhti kiihtyy, kiihtyy myös tiedon katoamisen vauhti. Verkkokulttuurin taltiointi on yhtä tärkeää kuin kenkien ja kolikoiden museointi.

Vesa Linja-aho

Kesäkuussa 2002 sen aikaisessa somessa – irkissä, keskustelufoorumeilla ja kahvipöydissä – kävi kova pöhinä nimettömänä julkaistusta kirjasta *Minne hävisivät Soneran rahat?* Tässä 190 048 merkin – eli yhdessä illassa luettavan tietokirjan – pituisessa tekstitiedostossa kerrottiin värikkäin sanankääntein Soneran eli nykyisen Telian myrskyisästä lähihistoriasta.

Tiivistelmänä: Valtion puhelinyhtiö Tele, joka omisti osan kotimaisesta lankapuhelinverkosta ja oli toinen suurista matkapuhelinoperaattoreista Suomessa, nimettiin Soneraksi 1998 ja listattiin pörssiin 1999. Firman pörssikurssia ajettiin ylöspäin kuin käärmettä pyssyyn hypettämällä kovia aikomuksia vailla liiketoiminnallisia näyttöjä. Sonera ei ollut ainoa tapaus laatuaan, vaan sama tauti vaivasi monia muitakin uuden median ja tietotekniikka-alan yrityksiä ja sijoittajia.

Kun kurssit lopulta romahtivat, puhuttiinkin *it-kuplan* puhkeamisesta.

Sonera-kirja

Kuplan puhjetessa Sonera, jonka myyntivoitoilla oli hurjimmista kuvitelmissa tarkoitus maksaa jopa valtionvelat pois, kaatui ruotsalaisen Telian syliin pilkkahinnalla. Pari kuukautta fuusioaikeista ilmoittamisen jälkeen, kesällä 2002, internetissä levisi tuntemattoman kirjoittajan – tai kirjoittajien – paljastuskirja, jossa esitettiin piikkäin sanankääntein paljastuksia Soneran rämäpäisestä johtamisesta.

Kirjan väitteiden todenperäisyydestä ei ole yksiselitteistä tietoa. Sekä tieteessä että arkielämässä yleinen periaate on, että todistustaakka on väitteen esittäjällä ja journalismissa tiedot tulee varmistaa vähintään kahdesta toisistaan riippumattomasta lähteestä. Ja mitä kummallisempi väite on, sitä vedenpitävämmät todisteet tarvitaan. Kirjassa esitetyistä hurjista väitteistä todeksi on osoittautunut ainakin tele-

tietojen urkinta, joka lopulta johdatti Sonera-kirjan ”pääpahiksen”, toimitusjohtaja **Jan-Erik Relanderin** tuomitsemiseen törkeästä viestintäsalaisuuden loukkauksesta. Koko jupakka toi tuomion usealle muullekin soneralaiselle.

Sen sijaan kirjan väitteistä hurjimmat kaipaisivat tuekseen muitakin todisteita kuin anonyymit väitteet. Näitä olivat

esimerkiksi se, että Venäjän mafiaa olisi käytetty erään äänekkäästi Soneran asiointia kommentoineen Soneran hallinnon jäsenen hiljentämiseen sekä se, että Relander olisi junailut Soneran rahoja itselleen ulkomaisten peiteyhtiöiden kautta.

Se, että kirjan väitteitä ei ole pystytty kumoamaan ja väitteistä osa on osoittautunut todeksi, kallistaa vaakaa siihen suuntaan, että kirjan kirjoittajat tietävät mistä puhuvat. Toisaalta kikka, jossa uskottavuutta tuovan tosiasiarypään sekaan ujutetaan muutama valhe, lienee yhtä vanha keksintö kuin valehtelukin.

Vaikka kirjan julkaisusta on jo 17 vuotta, se pulpahtelee edelleen pinnalle säännöllisin väliajoin: helmikuussa 2015 Helsingin Sanomat uutisoi, että kirjaa koskevaa Wikipedia-artikkelia on vaadittu poistettavaksi Googlen hakutulokista eurooppalaisen tietosuojalainsäädännön perusteella. Myöhemmin samana vuonna uutisoitiin Relanderille aikanaan korvauksia maksaneen vakuutusyhtiön riitelevän teleurintajupakkaan liittyvistä asianajokuluista Espoon käräjäoikeudessa.



Umtsi-um

Suurin yksittäinen ja samalla kuuluisin käänne Sonera-saagassa oli Relanderin päätös käyttää liki neljä miljardia euroa Saksan valtiolta ostettuun umts-toimilupaan. Koko investointi jouduttiin pian kirjaamaan tappioksi tilinpäätöksessä, ja Soneran utms-luvat olivat kaikkien huulilla. Muistan elävästi, kun radiojuontajat pohtivat ääneen, mistä tarkalleen ottaen mahtaa olla kysymys, koska umts ei tuo heille mieleen mitään muuta kuin vettä täynnä olevista kumisaappaista kuuluvat askeläänät. Ei liene sattumaa, että umts-tekniikan viimein tullessa käyttöön vuosikymmenen lopulla se brändättiin Suomessa 3g:ksi ja umts-termi haudattiin visusti. Vasta simkortin ostaminen ulkomaanreissulla 2010-luvun alussa palautti mieleen, että kyse on aivan samasta tekniikasta, kun nämä neljä kirjainta komeilivat mainostamassa nopeaa datasiirtoa.

Luin Sonera-kirjan kunnolla vasta tänä kesänä lomallani, vaikka tapaukseen liittyvälle uutisoinnille hihittelin jo parikymppisenä teekkarina. Kirjan lukemiseen ajoi tietynlainen nostalgian nälkä yhdistettynä sattumaan:

keväällä ilmestyi *The Intelligence Trap*-kirja. Siinä mainittiin yhtenä älykkäitä ihmisiä koskevana organisatorisen tyyperyyden riskinä taipumusta poskettomaan riskinottoon, mikä toi välittömästi mieleen Soneran ja it-kuplan. Googlasin mitä paljastusteokselle kuuluu nykyään ja luin sen vihdoinkin yhdeltä istumalta.

Tavallaan ymmärrän Relanderia: yhtiön valtio-omistaja oli asettanut kirjaimellisesti tolkuttomat kasvutavoitteet yrityksen johdolle optio-ohjelman kautta. Optiokannustin kannustaa kasvattamaan yrityksen pörssikurssia, joka ideaalimaailmassa korreloi yrityksen rahantekokyvyn ja kasvuodotusten kanssa. Toisin kun Nokia, joka valmisti ja myi matkapuhelimia ympäri maailmaa, Sonera oli valtion puhelinyhtiö, jonka bisnes perustui kotimaan lanka- ja matkapuhelinverkkoon. Vaikka jälkiviisastelu on tunnetusti helppoa, en keksi edes jälkikäteen Relanderin strategiaa parempaa keinoa odotuksiin vastaamiseen. It-kupla vain lässähti käsiin ennen firman myyntiä.

Nokiasta puheen ollen, firman noususta ja (lähes) tuhosta on julkaistu useita populaarikirjoja ja akateemisia tutkimuksia, pitkiä lehtijutuista puhumattakaan. Sonerasta vastaavat teokset loistavat poissaolollaan. Pelkästään Sonera-kirjan väitteiden perkaamisessa olisi aihetta kokonaiseksi tietokirjaksi. Tapahtumista on kohta 20 vuotta ja osa avainhenkilöistä jo eläkkeellä ja kenties valmiina avaamaan suunsa vähintään lähdesuojan turvin, joten tässä olisi Helsingin Sanomien kuukausiliitteelle tai Long Playlle journalistipalkintoa jaossa. Saksan umts-lupien osto on sentään päässyt esimerkiksi Markku Kuisman ja Pekka Seppäsen vuonna 2015 ilmestyneeseen *Suomen pahimmat bisnesmokat*-kirjaan.

Katoavainen data

Sonera-kirja löytyy verkosta vieläkin. Kirjasta julkaistiin 2003 myös painettu versio, josta oli hurjimmat väitteet sensuroitu pois. Kirjaan tutustuminen muistutti kuitenkin digitaalisen kulttuurin lyhyestä eliniästä ja helposta katoavaisuudesta. Kolumnoin aiheesta Skrollissa 2016.2, johon **Janne Sirén** viittasikin mainiossa BBS-historiakolumnissaan edellisessä Skrollin numerossa.

BBS:t ovat hyvä esimerkki ilmiöstä, joka katosi nopealla aikataululla, mutta jonka vaikutusta tietokonekulttuurille ei voi kiistää. Digimaailmassa moni hetki sitten itsestään selvä asia häviää, ennen kuin sen talteenottoon herätään. Aihe palasi mieleen, kun yritin selittää 10-vuotiaalle lapselleni, että nuoruudessani puhelimilla lähetettiin sellaisia merkkigrafiikka-animaatiotekstiviestejä, joita alaspäin skrollatessa (sic!) tapahtui jotain hassua – yleensä kaksimielistä. Paljon selittämistä helpompaa olisi ollut näyttäminen. Mutta kenellä näitä on tallessa? Taustakuvista ja soittoaänistä, joiden tilaamisesta vielä viime vuosikymmenellä muistan maksaneeni, sentään on tallessa lehtimainokset Kansalliskirjaston arkistossa. Soittoaäntä en muista tällä vuosikymmenellä vaihtaneeni kertaakaan. Taustakuvana on maksul-

"...digitaalisen kulttuurin katoavaisuus iskee päin naamaa aina, kun kertoo omille lapsilleen jostain tuoreehkosta mutta jo tyystin kadonneesta ilmiöstä."

lisen pikselimössölogon sijaan puhelimen omalla kameralla otettu eläinkuva, jollaista ei 2000-luvun alussa olisi saanut edes markkinoiden kalleimmilla digikameroilla.

Suomen ja maailman museot pulistelevat monenlaista puukauhua ja kenkää, mutta digitaalisen kulttuurin katoavaisuus iskee päin naamaa aina, kun kertoo omille lapsilleen jostain tuoreehkosta mutta jo tyystin kadonneesta ilmiöstä. Fyysisten vempelien säilöminen toimintakuntoisena on tärkeää sekin, mutta sisällön tallentaminen on oma haasteensa. Pikaisen kyselykierroksen perusteella esimerkiksi sosiaalisen median sisältöjen kattavasta tallentamisesta ei vastaa tällä hetkellä Suomessa mikään tahon.

Tietojärjestelmäasiantuntija **Petteri Veikkolainen** Kansalliskirjastosta (ks. myös Skrolli 2018.2) kertoo Skrollille, että some-aineistoista Kansalliskirjasto tallentaa erityisesti verkkoarkiston teemakeräyksissä Twitterin ja YouTubein sisältöjä. YouTubeista tallennetaan myös itse videotiedosto kommenttei-

neen, eli vaikka Google palveluineen yhtäkkiä pyyhkäistäisiin pois bittivaruudesta, historiantutkijoille jää käyttöön jonkinlainen otos YouTube-videoista. ”Some-palvelujen aineistojen tallettaminen on työlästä, ja useita merkittäviä kanavia jää tallettamatta, koska se on yksinkertaisesti todella hankalaa”, Veikkolainen vahvistaa.

Yksi haasteellinen tallennettava ovat mobiilipelit ja verkkopelit. 1990-luvun klassikkopelejä voi pelata emulaattoreilla ja ehjällä alkuperäisellä raudalla, mutta pääsevätkö mahdolliset lapsenlapseni rökittämään minut Clash Royalessa tai Fortnite:ssä kuten minä aikanaan rökitin isoisäni shakissa ja Monopolissa? Verkkopelit nimittäin yleensä katoavat siinä vaiheessa, kun palvelimen ylläpitokustannukset kasvavat pelin tuomaa kassavirtaa suuremmaksi. Kuka museoisi sen palvelinpään ohjelman? ”Kansallisen kulttuuriperinnön näkökulmasta erityisesti suomalaisten pelien tallettaminen olisi tärkeää. Oma arvaukseni on, että kotimaisten pelitalojen valmius – ja halu – tarjota tarkoitukseen sopiva versio ei koidu ongelmaksi, jos Kansalliskirjastolla olisi valmius kehittää järjestelmä tämän toteuttamiseksi. Suomessakin pelattavien ulkomaisten jättihittien tallettaminen on sitten toinen juttu”, jatkaa Veikkolainen.

Entä muu verkkokulttuuri? Kansalliskirjastolla on toki bottinsa, joka tallentaa verkkosivuja jälkipolville. Pelkän mekaanisen keräämisen lisäksi olisi tärkeää myös taustoittaa tapaukset. Vaikka Hesarin Nyt-liitteen yhteenvedotut kaikenkarvaisista someriidoista ärsyttävät monia, niillä on tärkeä merkitys myös kulttuurihistorian säilömisessä.

Museoalalla on peruskauraa, että pelkän esineen tallettamisen sijaan on tärkeää tallettaa myös taustatiedot esineestä: kuka sen on valmistanut, kuka sitä käytti ja mihin, ja mikä oli esineen merkitys aikalaisille laajemmassa kontekstissa? Tällaisten tietojen kerääminen on käsipeliä. Kansalliskirjaston verkkorobotin haaviin on satavarmasti tarttunut niin Sakolut:den paintaidetta kuin anonymien väsämiä *Spurdo spärde* -kuviakin, mutta ilman asiantuntevaa taustoitusta nämä eivät jälkipolville aukea. Pelimuseo meillä jo on Tampereella, pitäisikö sen kylkeen avata meemimuseo? Sellainen sopisi mainiosti mediamuseo Rupriikin yhteyteen, itsestäänselvine synergiaetui-

KIRJOITTAJAN TYÖKALUPAKKI

Tuotatko paljon tekstiä tietokoneella? Aiheuttaako viitteidenhallinta harmaita hiuksia? Esittelyssä kirjoittamisen ja viitteidenhallinnan tehoyökalut.

Teksti ja kuvat: Vesa Linja-aho

Termi kirjoitusohjelma tuntuu hieman erikoiselta – mihin nyt tarvitaan erilaista ohjelmaa kirjoittamiseen: useimmat ovat täysin tyytyväisiä Wordiin, Google Docsiin tai vaikka lempitekstieditoriinsa. Tietyntylaiseen kirjoitustyöhön voi olla hyödyllistä käyttää erillistä ammattikirjoittajille suunnattua sovellusta, jossa tekstieditori on yhdistetty tekstintuotantoa tukeviin sisällönhallintatoimintoihin ja lopullisen teoksen jäsentelyyn tarvittaviin toimintoihin.

Alan markkinajohtajassa Wordissa ominaisuudet riittävät aina matemaattisista kaavoista lähdeviitteiden hallintaan, mutta paras työkalu se ei välttämättä näihin toimintoihin ole. Esimerkiksi lähdeluettelon muotoilu- vaihtoehdot ovat rajoitetut ja vähänkin monimutkaisempien kaavojen syöttö hankalaa.

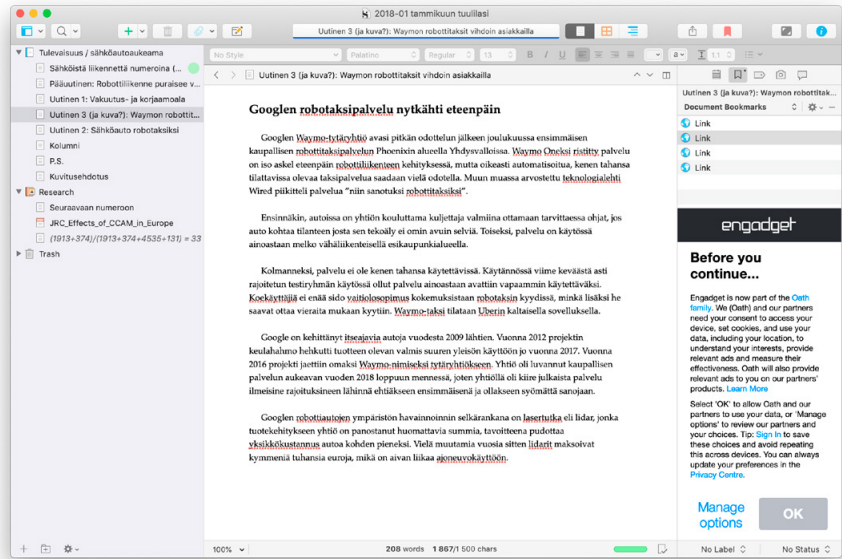
Yleisin vaihtoehto Wordille on Google Docs, joka kilpailee tarjoamalla toimivat versionhallinta- ja etenkin ryhmätuoto-ominaisuudet. Viitteiden hallinta onnistuu Docsissa erillisen liitännäisen avulla.

Nämä kaksi jättäviä eivät ole suinkaan ainoat laadukkaat työkalut kirjoittamiseen ja viitteiden hallintaan. Paljon kirjoitustyötä tekevien kannattaa vähintään kokeilla muutamaa muuta työkalua. Tässä artikkelissa esitellään kirjoitusohjelmat Scrivener, Ulysses ja IA Writer sekä viitteidenhallintaohjelmat Paperpile, Refworks, Papers, Zotero, Endnote ja Mendeley.

Kirjoitusohjelmat

Scrivener

Tunnetuin kirjoitustyöhön tarkoitettu ohjelmisto lienee vuodesta 2007 asti kehitetty Scrivener. Ohjelma on käytännössä kevyin ominaisuuksin



Scrivenerissä on monipuoliset ominaisuudet etenkin laajojen tekstikokonaisuuksien hallintaan.

varustettu rtf-tekstieditori, jossa on laajat sisällönhallintaominaisuudet ja miellyttävä käyttöliittymä. Kirjojen ja laajojen lehtijuttujen kirjoittamista helpottavat lukuisat ominaisuudet, kuten korkkitaulunäyttö eri tekstin osioille ja monipuoliset mahdollisuudet lisätä kommentteja, värikooditappiä ja merkkimäärätavoitteita eri tekstitiedoille.

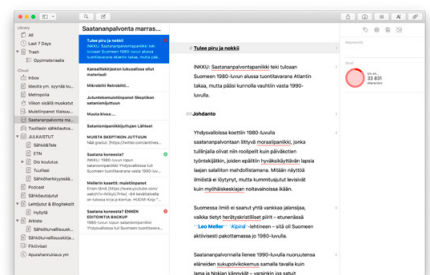
Olen kirjoittanut ohjelmalla yhden tietokirjan, ja kokemus oli positiivinen. Varsinkin jos sisältö on lähinnä tekstejä ja taulukoita ja projekti on laaja, Scrivenerin kanssa ei ainakaan häviä mitään. Etenkin eri lukujen pituuksien seuraaminen graafisesti oli yksinkertainen mutta hyödyllinen pikkuominaisuus.

Scrivenerin merkittävin huono puoli löytyy pikkuyksityiskohdasta: ohjelma pakottaa käyttämään wysiwyg-henkistä editoria, vaikka plain text -ominaisuutta on ohjelman tukifoorumilla toivottu jo 2008. Jos kirjoitusprosessi sisältää paljon lähdeoteuksista kopioimista, turhien muotoiluominaisuuksien mukana roikottaminen vain yksinkertaisesti ärsyttää.

Scrivener sisältää paljon kirjoittamistyötä helpottavia ominaisuuksia, kuten erilaisia editoritiloja: näytöltä saa piilotettua kaiken turhan, vieritysmoodin saa kirjoituskonemaiseksi (editoitava rivi korostettuna ja samalla kohtaa näytöllä) ja useamman dokumentin

tarkastelu sekä dokumentista toiseen hyppiminen onnistuu pikanäppäimillä helposti. Lyhyiden tekstien kirjoittamisessa Scriveneristä lienee lähinnä haittaa, mutta laajemmissa projekteissa kannattaa vähintään kokeilla alussa, olisiko Scriveneristä niiden suunnittelussa, jäsentelyssä ja kirjoittamisessa apua. Ohjelmaa voi koekäyttää 30 vuorokautta maksutta.

Jos kirjoittamista helpottavat editoriominaisuudet houkuttavat mutta tekstiä käsittelee mieluummin puhtaanä tekstinä ja hoitaa mahdolliset muotoilut Markdown-kielellä, kannattaa tutustua Ulyssesiin ja IA Writeriin.



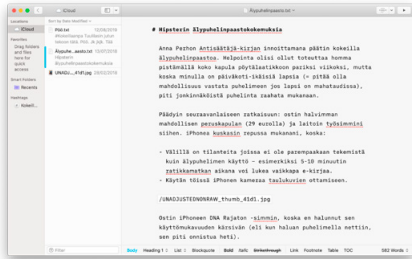
Jos tuotanto on pelkkää tekstiä ilman komentokenkejä, Ulysses on miellyttävä työkalu.

Ulysses

Viimeistellyllä käyttöliittymällä ja pomminvarmalla synkronoinnilla varustettu ohjelma on ollut pääkirjoitus työkaluni kesästä 2017 alkaen. Ohjelma on käytännössä Markdown-editori joka tallentaa dokumentit iCloud-tietokantaan. Tekstin lisäksi ohjelmaan

on helppo lisätä kuvia leikkaamalla ja liittämillä tai lisäämällä tiedostosta.

Eniten riesaa aiheuttava puute on taulukoiden puuttuminen. Yhden taulukon tekee tarvittaessa toisellakin ohjelmalla, mutta jos niitä on useampi, koko artikkelin kirjoittaa mieluummin vaikkapa Google Docsissa.



IA Writer tukee Markdown-taulukoita toisin kuin Ulysses, mutta kuvatuki on kehnempi.

IA Writer

Jos Ulysses tuntuu käteen sopivalta, kannattaa kokeilla myös IA Writeria – ja päin vastoin. Ohjelmat ovat hyvin saman tyyliisiä, ja erot löytyvät nyansseista. IA Writerin kuvien lisääminen on hieman kolhompia: kuva ei näy tekstin seassa esikatseltavana vaan ainoastaan tiedostonimenä. IA Writerin ehdoton vahvuus Ulyssesiin verrattuna on tuki taulukoille Markdown-kuvauskielillä.

Lähdeviitteet haltuun

Tekstinkäsittelyohjelmista ainakin Wordissa on jonkinlainen lähdeviitteiden hallintatoiminto. Pienehkön lähdeviitemäärän kanssa se toimii mainiosti, kuten myös silloin, jos lopulliseen teokseen vaadittavan lähdeluettelon muotoilu löytyy valmiiksi Wordin viittaustyylivalikoimasta.

Usein pienen lähdeviitemäärän kanssa pääsee helpoimmalla, kun suorittaa viitteidenhallinnan käsin. Tämä on erityisen helppoa silloin, jos käytetään nimi-vuosi (Möttönen 2018) -viittausjärjestelmää. Teknis-tieteellisillä aloilla suosituimpi on numeroviittausjärjestelmä [1], joka on kätevä etenkin silloin, kun viitattavana on paljon standardeja tai muita lähteitä, joilla ei ole selkeää henkilötekijää. Numeroviittausjärjestelmää käsin käyttäessä on helppo sotkeutua omaan näppäryyteensä, jos lisää uuden viitteen jo olemassa olevan tekstin alkupäähän ja numerointi menee sekaisin. Yksi nokkela tapa selättää tämä ongelma on nimetä viitteet itselle mukavalla tavalla, kuten vaikkapa [muovistandardi], ja sitten valmiissa dokumentissa korvata nämä numeroilla.

Jos viitteitä on useita kymmeniä tai – kuten etenkin tieteellisissä katsausartikkeleissa ja kirjoissa – useita satoja, viitteidenhallintaohjelmiston käyttö on enemmän kuin suositeltavaa. Tällöin lähdeluettelon ja viittausten muotoilu onnistuu napinpainalluksella. Monessa viitteidenhallintatyökalussa on myös toimiva hakuominaisuus, joka helpottaa tieteellisten artikkelien löytämistä. Etenkin katsauksia tehdessä hakutoiminnosta on iso apu. Toinen helpottava ominaisuus on automaattinen viitteiden muotoilu. Paperikirjoista viitetiedot joutuu edelleen syöttämään käsin, mutta tunnettujen tiedekustantajien artikkeleista kunnollinen viitteenhallintaohjelma tuottaa viitteen pelkän linkin syöttämällä tai raahaamalla pdf-tiedoston ohjelmaan.

Paperpile

Jos käytät jo Google Docsia, Paperpilen käyttöönotto on mutkatonta: kirjautu Paperpileen Google-tunnuksella, asenna Chrome-lisäosa ja ei kun ajelemaan. Lisäosakaan ei ole pakollinen, mutta se helpottaa viitteiden tuontia tiedekustantajien verkkosivuilta.

Omalle koneelle ei tarvitse mainitun Chrome-palikan lisäksi asentaa mitään: homma toimii pilvessä, eikä

Kokeile ja katso tykkätkö

Kirjoitusohjelmien paremmuusjärjestykseen laittaminen on kuin vertailisi Ferraria ja Hummeria. Paras vaihtoehto onkin kokeilla eri ohjelmia ja katsoa, kolahtaako.

Jos artikkeli on vain muutaman tuhannen merkin pituinen, on käytännössä sama, millä työkalulla sen kirjoittaa. Oman henkilökohtaisen tiedonhallinnan kannalta tosin on kätevää, jos kaikki kirjoitusprojektit ovat samassa sovelluksessa. Esimerkiksi Ulyssesistä löytää nopeasti yhdellä nopealla sanahauulla kaikki vanhat tiettyä teemaa koskevat artikkelinsa. Tämä on kätevä ominaisuus varsinkin vakiopalstoja kynäileville: olen kerran vahingossa meinannut kirjoittaa lähes samansisältöisen kolumnin kuin olin reilu vuosi sitten kirjoittanut vieläpä samaan lehteen. Jostain syystä tuotos ei ollut tallentunut pitkäkestoiseen muistiini.

Jos taas kirjoitustyöhön liittyy paljon erimuotoista oheismateriaalia, ei ole lainkaan huono idea käyttää perinteistä tietokoneen tiedostojärjestelmää. Tämä on tosin huomioitu myös Scrivenerissä: kirjoitusprojekteissa on oma kansio minkämuotoisille tiedostoille tahansa.

Hieman erikoisempaan kulmana voi mainita, että ohjelmaan saattaa myös kyllästyä. Olen käyttänyt Ulyssesta pääkirjoitusohjelmanani kesästä 2017 alkaen, ja välillä on virkistävää kirjoittaa vaihtelun vuoksi Google Docsissa. Sen kiistaton etu on, että artikkelia voi työstää vaikka jossain työpaikan atluokan perällä tai työkoneella: Ulysses toimii vain Applen tuotteilla.

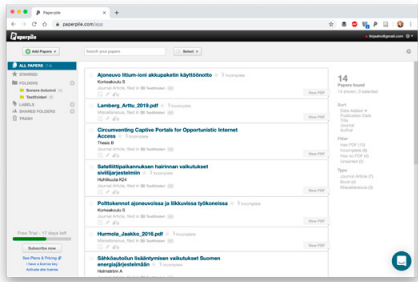
Hyvästä kirjoitus- tai viitteidenhallintaohjelmasta kannattaa myös maksaa. Kaksinumeroinen vuosimaksu takaa, että ohjelmaa kehitetään eteenpäin. Opiskelijalle summa on iso, ammattikirjoittajalle säläkuluja.

Yksi seikka ylitse muiden puoltaa Wordin käyttöä: lähes kaikki muut-

kin käyttävät sitä. Scriveneristä saa ulos suurin piirtein minkä muotoisen tiedoston tahansa, mutta kustannustoimittaja haluaa yleensä käsikirjoituksen Word-muodossa. Kun sitten kustannustoimittajan punakynä heiluu Wordissa, kuka jaksaa tai haluaa enää siirtää samoja muutoksia Scriveneriin? Tuskinpa kukaan. Se, haittaako tämä, on sitten asia erikseen. On täysin toimiva tapa tehdä ensimmäinen käsikirjoitus Scrivenerillä ja sitten jatkohieronta Wordissa.

Eri ohjelmia voi myös yhdistellä. Esimerkiksi tehtävälistasovelluksista itse käytän ideoiden ja isompien tehtävien hallintaan Thingsiä ja päivittäisen pikkusälän hallintaan Swipesiä. Istu siis tylsänä sadepäivänä alas, leiki eri kirjoitusohjelmilla ja etsi itsellesi toimiva yhdistelmä.

Jos artikkelia varten kokeiluista ohjelmistoista pitäisi valita voittajayhdistelmä, se olisi Google Docs ja Paperpile. Yhdistelmä toimii ja käytön oppii nopeasti.



Paperpile hallitsee viitteitä saumattomasti Google Docsin, Chromen ja Google Driven kanssa.

tarvitse esimerkiksi jännittää, toimii-ko synkronointi vai ei. Chromen saa käyttäjärjestelmään kuin käyttöjärjestelmään. Artikkelia voi työstää ja lähdeluetteloa päivittää niin kotona kuin töissäkin.

Ohjelmassa on laaja sisäänrakennettu haku, jolla voi etsiä artikkeleita eri tietokannoista. Hakutulokset näytetään löydettyjen artikkelien viittausten määrän mukaan laskevassa järjestyksessä, eikä tätä voi muuttaa asetuksista. Uuden tutkimuksen etsimisessä olisi kätevää, jos tulokset saisi myös aikajärjestyksessä. Hakutulokset saa kätevästi lisättyä omaan kirjastoon.

Artikkeleita voi tuoda ohjelmaan hakutoiminnon lisäksi vetämällä ja pudottamalla pdf:n suoraan kiintolevyltä. Jos kyseessä on vaikkapa paperikirja, viitetiedot voi luonnollisesti syöttää käsin joko kenttä kerrallaan tai tuoda RIS- tai BibTeX-muodossa. Viitetietojen käsin syöttämiseen on harvoin tarvetta, koska esimerkiksi suomalaisten kirjojen viitetiedot saa kätevästi tuotua Finna-palvelusta.

Oma lähdekokoelma on käytettävissä kaikilla Chrome-selaimilla, joihin kirjautuu sisään. Jos haluaa pelkkien viitteiden lisäksi synkronoida myös pdf:t, pitää kytkeä erikseen päälle Google Drive -synkronointi.

Toimiva synkronointi on kuin raskaus

Jos käyttää ohjelmaa usealla laitteella, olennainen ominaisuus on toimiva synkronointi. Ulysses on esimerkki tästä. iCloud-synkronoinnin kanssa ei ole ollut koskaan ongelmia. Lehtijuttua voi kirjoittaa tietokoneella, ja jos kesken kadulla kävelyn saa hyvän juttuun liittyvän oivalluksen, sen voi lisätä suoraan juttuun puhelimella ja viimeistellä sitten koneella.

Jos kyse on vähänkin tunnetumman kustantajan tiedeartikkelista, lähdeviitteen saa lisättyä pikanäppäinkomennolla suoraan url:n perusteella Google Docs -asiakirjaan.

Paperpilea on mukava käyttää. Pieni käytettävyyksiin tulee siitä, että kun lisää uuden viitteen, se muotoutuu aina Harvard-muodossa (Peloton 2003), vaikka viitteiden ja lähdeluettelon muotoiluksi olisi jo valinnut jotain ihan muuta. Tätä ei saa muutettua, ei edes valitsemalla uuden asiakirjan vakioviit-taustyyliksi jotain muuta. Jotta viitteet saa taas haluamaansa muotoon, pitää valita käsin Format citations, minkä jälkeen lisäosa ruksuttaa viitteen sekuntia ja kaikki on taas kunnossa.

Jos Google Docsia käyttää muutenkin artikkelin kirjoittamiseen, Paperpile tuntuu ykkösvalinnalta. Google Docs on suosittu alusta varsinkin jos tekstiä työstää useampi kirjoittaja eri puolilla maailmaa eikä teksti sisällä liitaa matemaattisia kaavoja, jolloin valinta kallistuu LaTeXin puolelle. Kun artikkeli on valmis, Google Docs -asiakirja on helppo viedä vaikkapa julkaisusarjan omaan Word-pohjaan.

Paperpilen viitteidenhallinta toimii mutkattomasti myös useamman kirjoittajan kanssa, kunhan muistaa lisätä viitteet jaettuun kansioon. Vaikka Docs toimii ilman verkkoyhteyttä Chromeselaimesa, Paperpile vaatii verkkoyhteyden. Jos verkkoyhteyttä ei ole, uusia viitteitä ei voi lisätä.

Refworks

Refworks on kenties tunnetuin selain-pohjainen viitteidenhallintajärjestelmä. Ohjelma on organisaatiohinnoiteltu ja käytettävissä lähes kaikissa suomalaisissa yliopistoissa ja ammattikorkeakouluissa. Käyttöoikeuteen riittää oikea sähköpostiosoite domain-osa.

Scrivenerissä synkronointi Windows- ja Mac-versioiden kanssa toimii muuten kauniisti, mutta välillä fontti vaihtui synkronoinnin seurauksena. Tällainen sinänsä harmiton juttu – kustantaja tietenkin valitsee fontin lopulliseen teokseen – sai lopettamaan synkronoinnin kanssa leikkimisen kokonaan, koska fonttien sotkeminen herätti epäilyksen siitä, mitä muuta synkronointi mahdollisesti hukkaa.

Aivan kuten ihminen ei voi olla ”melkein raskaana” tai ”vähän ras-

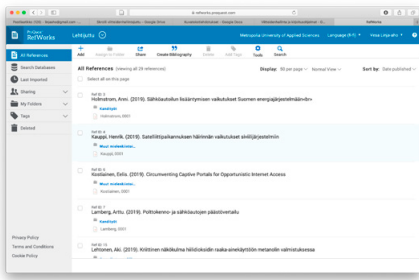
Perinteistä ja toimivaa: LaTeX ja Vim

Viime vuosikymmenellä Teknillisessä korkeakouluissa oli kahdenlaisia professoreita: niitä joiden oppimateriaalin ulkoasu oli tehty vähän sinne päin ja niitä joiden oppimateriaali oli tehty LaTeXilla. LaTeX on tekstinladontaohjelmisto, jolle annetaan syötteenä lähdekoodi, ja tuloksena saadaan (tavallisesti) pdf-tiedosto. LaTeXin valtti on viimeistely ja ammattimainen lopputuloksen ulkoasu sekä monipuolisuus: ohjelmistolla onnistuu käytännössä katsoen kaikki matemaattisista kaavoista lähdeluettelon, virtapiirikaavioiden, näyttävien kuvaajien ja jopa hakusanaston hallintaan. LaTeX on saatavilla myös kaikille äkkiä mieleen tuleville käyttöjärjestelmille. Tekstiä voi muokata lempitekstieditorillaan, itse käytän Vimiä (ks. Skrolli 2015.3). Jos perusominaisuudet eivät riitä, LaTeXia voi laajentaa lukuisilla lisäosilla. Esimerkiksi viitteidenhallintaa varten löytyy BibTeX, jos jo melko kattavat perusominaisuudet eivät riitä.

LaTeXin käyttö vaihtelee aloittain: tietojenkäsittelytieteilijöille ja matemaatikoille ohjelma tuntuu olevan itsestäänselvyys – muualla käyttö vaihtelee. Jos kollegasi ei jo käytä LaTeXia, hän ei välttämättä enää jaksa opetella. Harmi, koska monesti kannattaisi. Itsellä jako menee niin, että jos teoksessa on enemmän kuin muutama matemaattinen kaava, teen sen LaTeXilla.

Jos projekti on vähänkin laajempi, kannattaa käyttää ohessa jotain sopivaa versionhallintaa.

kaana” vaan hän on joko raskaana tai sitten ei, ohjelmassa on joko toimiva synkronointi tai sitten sitä ei ole. Synkronoinnin satunnainenkin sekoilu vie luottamuksen koko touthusta ja tekee toiminnosta käyttökelvottoman. Kirjoitustyö on välillä stressaavaa, eikä tämä stressi kaipaa tuekseen lisästressiä siitä, että kaikki uudet muutokset tulivat varmasti tallennettua.



Refworks on vanha mutta laajasti käytetty selainpohjainen viitteidenhallintaohjelma.

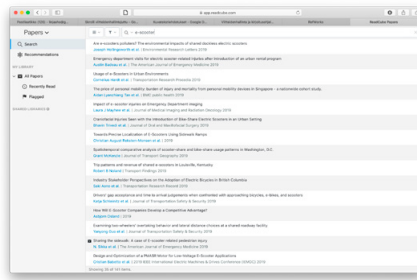
Käyttöliittymä – ohjelmiston dokumentaatiosta puhumattakaan – tuo tuulahduksen 1990-luvulta. Arvaus ei osu pahasti pieleen, sillä ohjelma on lanseerattu vuonna 2001. Noilta ajoilta lienee myös tietokantaohjelmointi: lähteen tietojen muokkaamisen jälkeen saa välillä odotella sekuntikaupalla, että jotain tapahtuu. Myös lähteen siirto toiseen kansioon aiheuttaa muutaman sekunnin pyörivän rantapallon katselutuokion.

Koeponnistin ohjelmaa laatiessani katsausta Aalto-yliopistossa sähköautoista tehdyistä opinnäytetöistä. Halusin lähdeluetteloon ehdottomasti näkyviin, onko kyseessä väitöskirja, diplomityö vai kandidaatintyö. Tämä ei onnistunut edes tuotteen omalla tyylieditorilla, vaan *Degree type* loisti poissaolollaan. Lopuksi päädyin ”puukotusratkaisuun” ja kirjoitin työn tyyppin instituuttikenttään. Ei eleganttia, mutta toimii – varsinkin nyt kun kaikki työt olivat samasta opinahjosta. Tässä vaiheessa tuli ikävä ihan käsin tehtävää viitteenhallintaa – tai LaTeXia, jossa puukotus olisi ollut nopea tehdä vaikkapa Vimin etsi ja korvaa -toiminolla.

Kahden tunnin tappelun jälkeen – joka ei mennyt hukkaan, opin paljon muun muassa CSL-lähteidenmuotoilukielestä (citationstyles.org) ja sain ongelman syyn jäljitettyä – jätin bugiraportin ongelmasta. Vastaukseksi sain, että ongelma on tiedossa ja ratkaisua etsitään. Keskustelupalstatietojen mukaan on etsitty kauan. Herää kysymys, onko ohjelman kehitystyö vähän niin ja näin, kun instituuttilaajilta valuu jatkuva kassavirta kehittäjän laariin. Omasta pussista ni en moisesta tekeleestä maksaisi.

ReadCube Papers

Ohjelma on Applen tuotteille kehitettyjen Papers 2:n ja Papers 3:n manttelinperijä. Etenkin Papers 3 oli surullisen kuuluisa siitä, että se saattoi hukata pdf-dokumentteja synkronoinnin aikana – tästä löytyy sekä omakohtaista

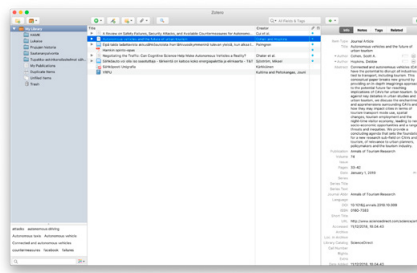


Papers on hieman keskeneräisen oloinen käytettävyydeltään, mutta sisältää mutkattoman hakutoiminnon.

kokemusta että tapauksia keskustelufoorumeilla. Readcube Papersin kanssa en saanut ongelmaa toistettua.

Käyttöliittymä on siisti, ja Paperpilen tapaan ohjelmassa on sisäänrakennettu kattava hakutoiminto. Selkeä etu Paperpilen hakutoimintoon verrattuna on se, että hakutulokset voi järjestää haluamallaan tavalla. Esimerkiksi uuden tutkimuksen seuranta on helppoa lajittelemalla tulokset aikajärjestykseen.

Viittausten käyttö onnistuu valitettavasti tällä hetkellä vain Microsoft Wordissa lisäosan avulla. Papers 3:sta tuttu Magic citations -toiminto (jolla viitteiden lisääminen onnistui lähes ohjelmaan kuin ohjelmaan) on valitettavasti poistunut keskuudestamme. Tietty keskeneräisyyden tuntu saa kiertämään ohjelman kaukaa, vaikka helppo hakutoiminto houkuttaakin.



Avoimeen lähdekoodiin perustuva Zotero toimii yleisimmillä käyttöjärjestelmillä.

Zotero

Vuodesta 2006 kehitetty Zotero on avoimeen lähdekoodiin perustuva viitteidenhallintaohjelmisto, johon nojaa muun muassa Wikipedian sitaattiautomaattikka: siististi muotoiltujen lähdeviitteiden lisääminen helpottuu, kun käyttäjän ei tarvitse syöttää käsin kirjoittajaa, median nimeä ja julkaisuajankohtaa, vaan Zotero generoi ne suoraan linkin perusteella. Tämä Zoteron kirjallisuusluettelomoottori on ilman rekisteröitymistä kaikkien käytössä osoitteessa zbl.org. Sivus-

Musta hevonen: Mediawiki

Jos vertailee eri ympäristöihin kirjoittamista, toimivin kompromissi löytyy Wikipediasta eli täsmällisemmin sanottuna Mediawiki-ohjelmistosta. Ensinnäkin, sisältöä voi tuottaa joko kirjoittamalla koodia tai käyttämällä wysiwyg-editoria, ja näiden välillä voi vaihdella lennosta, ilman että mitään menee rikki. Toiseksi, tekstin historiaa on helppo tarkastella ja vertailla. Kolmanneksi, ohjelmisto tukee LaTeX-muotoisia kaavoja ilman sarvia ja hampaita. Ja neljänneksi: en ole missään nähnyt yhtä toimivaa lähdeviitteiden hallintajärjestelmää. Viitteiden lisääminen ja uudelleenkäyttö on helppoa.

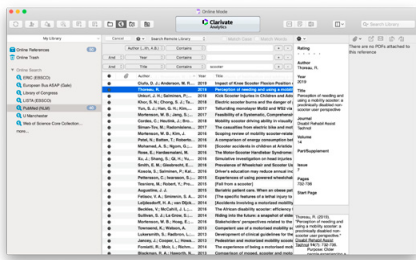
Miksi? Koska Wikipediaa toimittetaan vapaaehtoisvoimin, on tärkeää, että käytössä on helppokäyttöinen työkalu. Kun taustalla on suuri säätiö, myös resurssipuoli on kunnossa ja ohjelmiston kehitys jatkuu. Mutta kuinka moni jaksaa asentaa Mediawikin lokaalisti?

tolle voi syöttää url-osoitteita, DOI-tunnisteita tai ISBN-numeroita, joiden perusteella sivusto generoi lähdeluettelon halutulla tyyllillä.

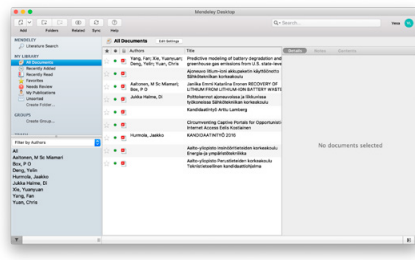
Zoteron sujuva käyttö viitteidenhallintaan vaatii käytännössä työpöytäohjelman asennuksen, mikä ei yleensä ole ongelma, koska Zotero tukee niin Windowsia, macOS:ää kuin Linuxiakin. Lisäosien avulla viitteidenhallinta sujuu niin Wordissa kuin Google Docsissa – tosin jälkimmäisessä integraatio on toteutettu jotenkin päälleliimatun oloisesti, kun työpöytäsovelluksen on oltava auki, jotta viitteitä pääsee muokkaamaan, viiveen kanssa tietenkin.

Endnote

Endnote on joukon veteraani: ensimmäinen versio julkaistiin vuonna 1988. Historian painolasti näkyy myös sekavassa käyttöliittymässä, joka tuo mieleen iPad-sukupolven lääkäreiden kiroamat terveydenhuollon tietojärjestelmät. Epäilyttäviä viboja aiheutti myös macOS:n varoitus ohjelman 32-bittisyydestä ja siitä, että ohjelma ei välttämättä toimi macOS:n seuraavassa versiossa. Ohjelman verkkosivuilla selitellään



Endnoten käyttöliittymä on hieman sekava – 1980-luvun painolastia?



Kustannusjätti Elsevierin kehittämä Mendeley toimii niin selaimessa kuin työpöydälläkin.

vuolaasti, että tämä kyllä korjataan tulevissa versioissa.

Endnote tekee saman minkä muutkin viitteidenhallintaohjelmat: lähdeluettelon saa koottua ja Word-liitännäisenkin toimii mainiosti. Ohjelmaa ei vain ole mukava käyttää. Ainoana positiivisena asiana voi mainita kattavan ja monipuolisen hakutoiminnon, jonka kautta pääsee käsiksi 432 avoimeen ja suljettuun artikkeli- ja kirjas-tietokantaan.

Mendeley

Mendeley on tiedekustantajajätti Elsevierin viitteidenhallintasovellus, jonka perusversio on maksuton, mutta asiakirjojen tallentamiselle voi Zoteron tapaan ostaa lisätilaa kuukausimaksulla. Ison talon isot resurssit näkyvät työn jäljessä: ohjelma toimii sujuvasti Windowsilla, Macilla ja pilvessä, ja synkronointi kaikkien kolmen välillä toimii saumattomasti.

Viittaamiseen löytyy Word-lisäosa, joka julistaa olevansa vasta beetaversio. Lisäosa toimii muuten moitteettomasti, mutta fonttikoko loikkaa suuremmaksi kuin leipätekstissä. Pientä viilaamista odotellessa...

Mitä yhteistä, mitä eroa?

Käytännössä kaikki viitteidenhallintaohjelmistot mainostavat tuhansien viittaus-tyylien valikoimaansa. Tässä on tehty järkevää yhteistyötä: viittaus- ja lähdeluettelotyylien määrittelyyn on kehitetty CSL-kuvauskieli, jolla kirjoitetut viittaus-tyylit on kerätty projektin sivustolle.

Viitteidenhallintaohjelmistot tarjoavat vähintään Word-lisäosan ja suurin osa myös Google Docs -tuen. Poikkeuksen Word-tukeen muodostaa alun perin Google Docsia varten kehitetty Paperpile, jossa siinäkin on Word-tuki jo beetatestivaiheessa.

Entäpä jos haluat käyttää viitteidenhallintaohjelmaa jossain harvinaisem-

Puuttuiko jotain?

Kun hakemalla hakee, työkaluja löytyy muitakin. Jäikö jokin ehdoton helmi käsittelemättä? Kerro se Skrollin Facebook-sivulla [facebook.com/skrollilehti](https://www.facebook.com/skrollilehti) tai lähetä postia toimitus@skrolli.fi.

massa kirjoitustyökalussa, kuten Scrivenerissä tai Ulyssesissa? Tämä onkin jo sellainen soppa joka vaatisi oman artikkelinsa. Lyhyesti: joillain yhdistelmillä onnistuu, joillain ei, ja niillä joilla onnistuu, homma voi olla yhtä hauskaa kuin hiekan pureskelu. Yksi tapa on tietenkin ylläpitää lähdeluetteloa erikseen esimerkiksi Zoterossa ja hoitaa viittaaminen ihan manuaalisesti vaikkapa Scrivenerissä. Nimi-vuosilukuviiteillä tämä onnistuu, numeroviittausjärjestelmässä taas tähän lähestymistapaan tuskin haluaa koskea pitkällä tikullakaan.

Paljon vaivaa säästävää automaattinen viitetietojen luominen asiakirjan perusteella toimii yhtä surkeasti kaikissa ohjelmissa, jos kyse on vaikkapa kotimaisista oppinäytetöistä tai vaikka suomalaisen kirjan tunnistamisesta isbn-tunnuksen perusteella. Kansainvälisten tiedekustantajien vertaisarvioituid artikkelit taas tunnustuvat oikein ohjelmalla kuin ohjelmalla. 🐼

	Paperpile	Refworks	Papers	Zotero	Endnote	Mendeley
Alusta	Google Drive	Verkkoselain	Verkkoselain, iOS, Android	Windows, Mac, Linux	Windows, Mac	Windows, Mac, verkkoselain
Tuetut alustat lisäosille	Chrome	Word, Google Docs	Word, Chrome, Firefox, Edge	Word, LibreOffice, Google Docs, Chrome	Word	IE, Chrome, Firefox, Safari, Word
Hinta	35,88 \$ (akateeminen lisenssi) tai 118,88 \$ (yrityskäyttö)	Organisaatiohinnoittelu	5 \$/kk (tutkijat), 10 \$/kk (yrityskäyttö), myös organisaatiohinnoittelu	Ilmainen, lisälevytilaa ostettavissa 20–120 \$/vuosi	249,95 \$	Ilmainen, lisälevytilaa ostettavissa 55-165 \$/vuosi
Muuta			Windows/Mac-työpöytäversio beetavaiheessa	Perustuu avoimeen lähdekoodiin.		

	Scrivener	Ulysses	IA Writer	Google Docs
Alusta	Windows, Mac, iOS	Mac, iOS	Windows, Mac, iOS, Android	Verkkoselain
Hinta	49 € (Windows), 53 € (Mac), 85 € (molemmat), 20 € (iOS)	39,99 €/vuosi	19,99 € (Windows), 29,99 € (Mac), 8,99 \$ (iOS)	Ilmainen, lisälevytilaa ostettavissa
Muuta			Android-versio on toistaiseksi ilmainen kokeiluversio	



OSA III

Sarja jatkuu Kuinka rakensin tietokoneen 1960-luvulla

Skrollin juttusarjassa ohjelmistosuunnittelija Joseph Watson kertoo, millaista oli rakentaa oma tietokone aikana ennen mikroprosessoria.

Teksti ja kuvat: Joseph Watson
Käännös: Mikko Heinonen

Viime kerralla tarina jäi siihen, että olin asentanut teräskaappiin itse rakentamani tietokoneen. Asuin tuolloin Yhdysvaltain ilmavoimien tukikohdassa Kaliforniassa. Tietokoneen näytöksi olin asentanut muutaman rivin hehku-lamppuja, tulostuslaitteeksi sähkökirjoituskoneen ja näppäimistöksi näppäinpuhelimen.

Tietokoneeni otti vastaan ja tulosti oktaalilukuja. Nyt kerron, miten kone saatiin laskemaan ja muistamaan niitä.

Vuonohjaus

Tietokoneessa oli kaksi eri komentoa ohjelman vuon hallintaan. JMP-komento siirsi ohjauksen ehdoitta tiettyyn

osoitteeseen. JPC-komento siirsi ohjauksen tiettyyn osoitteeseen vain, jos ehto täyttyi suoritushetkellä. Tässä on ehdottomasti yksi tietokoneen suunnittelun heikkouksia. JPC:n testaama ehto oli joko AC on erisuuri kuin 0 tai jokin AC:n biteistä on yhtä kuin 1 (en tosin enää muista, mikä bitti). Testattava ehto vaihdettiin kytkimellä. Tämä lähestymistapa tarkoitti, että tiettyssä ohjelmassa saattoi olla mahdollisia hyppyjä vain toisella näistä ehdoista.

Alirutiinit

Koneessa ei ollut pinoa. Koska CALL-komentoa ei ollut, JMP oli ainoa tapa siirtyä alirutiiniin. Ongelmana olikin, miten alirutiini osaisi palata sitä kutsuvaan rutiiniin. Se toteutettiin nykyisiin harvemmin käytetyllä tavalla: ohjelma muokkasi itseään.



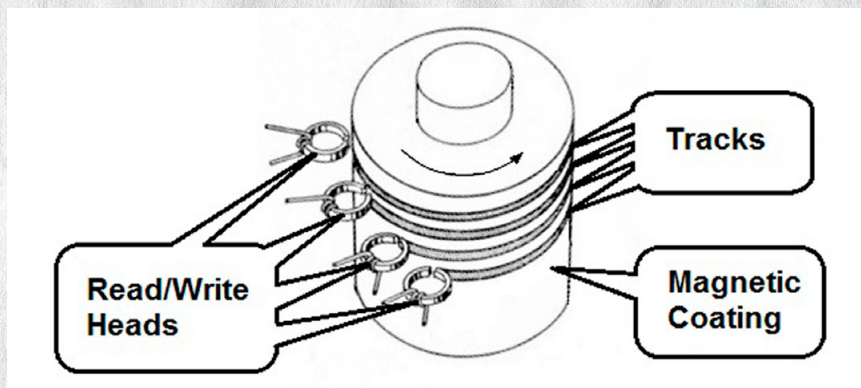
Joseph Watson liittyi Skrolli-kirjoittajien joukkoon kansainvälisen *Skrolli International Edition* -lehtiprojektimme yhteydessä. Tarina hänen myöhemmästä mikroprosessoria käyttäneestä mikrotietokoneestaan on julkaistu Skrollin numerossa 2016.4 (pdf-lehti: skrolli.fi/numerot).

Ennen kutsurutiinia oli hyppy alirutiiniin, joka latsi AC:hen paluuseen käytettävän JMP-komennon. Alirutiinin ensimmäiset käskyt tallensivat AC:n sisällön (JMP-komennon) osoitteeseen, johon piti palata. Alirutiinin suoritettua tehtävänsä se suoritti tallennetun

JMP-komennon ja hallinta palasi kutsurutiinille. Rekursiivinen alirutiinien kutsuminen ei ollut mahdollista ilman todella monimutkaista järjestelyä.

Varhaisia edistysaskelia

Ensimmäiseksi keskityin saamaan syöttö-, tulostus- ja laskuoperaatiot toimintaan. Koska toimintojen käynnistämiseen ei ollut käytettävissä ohjelmaa, asensin logiikkapakettin alle kolme painiketta, joista sai käynnistettyä ADD-, SUB- ja MUL- operaatiot, sekä muualle pari tilapäistä painiketta, joista käynnistyivät INPUT- ja TYPE- operaatiot. Noin kolmen kuukauden kuluttua nämä viisi toimintoa oli johdotettu ja ne toimivat oikein. Ohjelman tallentamiseen ei kuitenkaan ollut muistia, joten laite ei kyennyt tämän kummempaan vielä pitkään aikaan.



Magneettirummun toimintaperiaate.

Kertolaskusta

Ymmärsin, että kyky siirtää binäärilukua oikeaan tai vasempaan oli tärkeä. Kertomalla kahden potenssilla (ts. 2, 4, 8, 16...) sai aikaan siirron vasempaan, mutta oikeaan siirto ei onnistunut. Kaipasin jotakin tapaa sen toteuttamiseen.

Halusin myös keksiä tavan säilyttää kertolaskun tuloksen ylempi osa. PH-rekisterin lisääminen tallentamaan kertolaskun tuloksen ylempät bitit osoittautui osaratkaisuksi molempiin pulmiin, mutta vain, jos keksisin tavan palauttaa PH-rekisterin sisältö ohjelman käyttöön.

Tietokoneeni käyttämä alkuperäinen kertolaskumenetelmä näkyy seuraavassa vuokaaviossa (pois lukien varjostettu osa). Se alkaa kahdesta kerrottavasta luvusta, joista toinen on AC-rekisterissä ja toinen (MUL-käskyn operandi) on MD-rekisterissä. Laskun jälkeen tulon alempi osa on AC:ssä ja ylempi osa PH:ssa.

Saatuani tämän toimimaan aloin etsiä helppoa tapaa palauttaa PH-rekisterin arvo. Keksinkin sen lisäämällä logiikkaa, joka vastaa vuokaavion varjostettua osiota. Paremmankin testiehdon olisi ehkä voinut keksiä, mutta päädyin käyttämään tätä. Päätelin, että jos ohjelmoin nollalla kertomisen, tulos on aina 0, jolloin tällainen kertolasku on turha. Siksi nollalla kertomisen kohdalla vuoka-

vion varjostettu osio ohittaa PH:n tyhjennyksen. Muun vuon seuraaminen loppuun asti kertoo, että tämän muokatun prosessin kokonaisvaikutus on, että PH:n (aiemman tulon ylempi osa) sisältö siirretään AC:hen, josta ohjelma voi käyttää sitä.

Mahdollisuus palauttaa tulon ylempät bitit mahdollistaa myös oikealle siirron. Numeron kertominen 512:lla ja sitten 0:lla, jolloin PH-rekisterin arvo palautetaan, saa alkuperäisen numeron siirtymään 1 bitin oikealle. Kertominen 256:lla ja sitten 0:lla siirtää alkuperäistä numeroa 2 bittiä ja niin edelleen.

Muistitaistelu

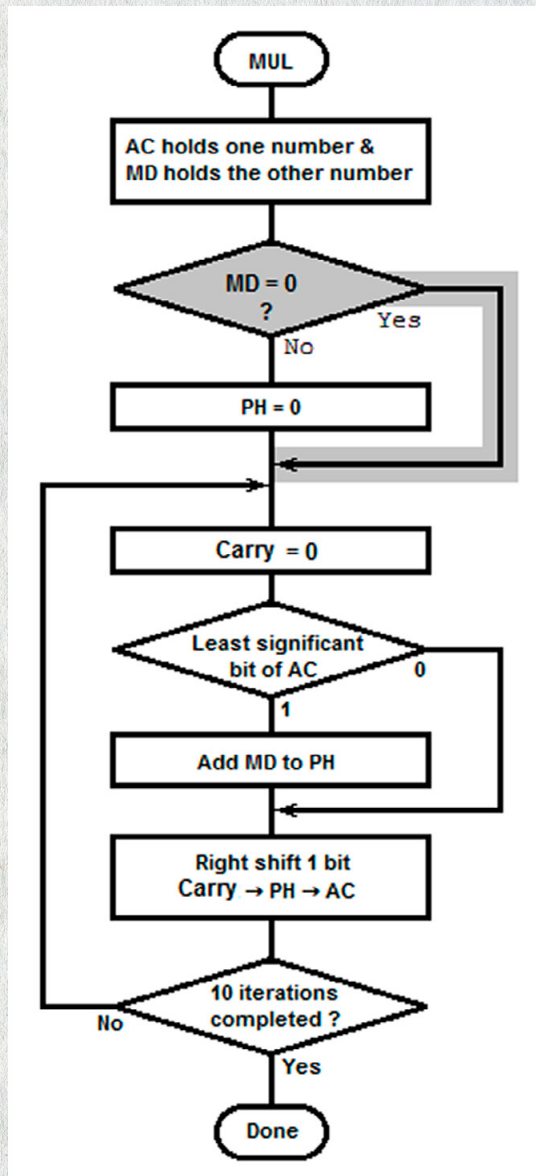
Tietokoneen täysi toiminta edellytti muistia. Käskyjen hakuoperaatiot sekä operandien luku- ja tallennusoperaatiot vaativat, että muistijärjestelmä voi lukea tietoja luotettavasta tallennusvälineestä tai kirjoittaa niitä. Mitä tallennusvälinettä voisin käyttää? Yritin rakentaa koneelle muistijärjestelmää monilla eri tavoilla.

Ilmavoimien koulutuksessa oli kerrottu ferriittirengasmuistista, mutta tätä muistityyppiä ei löytynyt ylimääliikkeistä.

Löysin elektroniikkaylijäämäliikkeestä magneettisen rumpumuistin ja ostin sen.

Magneettisessa rumpumuistissa on pyörivä metallinen (yleensä alumiininen) sylinteri, joka on pinnoitettu ulkopuolelta helposti magnetoituvala materiaalilla. Sylinterin pyöriessä luku- ja kirjoituspää, samanlaiset kuin kasettinauhureissa käytetyt, tallentavat tai toistavat tietoa magneettipinnalta.

Ostamani magneettirumpu oli suurikokoinen laite, joka painoi arviolta 15 kiloa. Se sisälsi myös moottorin. Rummun eri puolilla oli luku- ja kir-



Vuokaavio 10 x 10 bitin kertolaskulle.

joituspäitä, joilla pinnassa olevia raitoja käsiteltiin.

Kun yritin käynnistää rumpua, kuulin kyllä moottorin hurisevan, mutta rumpu ei pyörinyt. Irrottaessani lukupään näin, että sisällä olevassa magneettisessa materiaalissa oli rautaoksidipinnoite. Sen sävy oli samanlainen kuin ääninauhassa. Luku- ja kirjoituspäiden ei tulisi koskaan koskettaa magneettirummun pintaa, vaan ne pitää säätää hyvin lähelle sitä.

Näin, että lukupää oli joko säädetty liian pitkälle sisään tai sitä oli ajettu niin suurella virralla, että pää oli ylikuumentunut ja laajentunut magneettipinnan sisään. Joka tapauksessa oli selvää, että tuon raidan magneettipinta oli vioittunut.

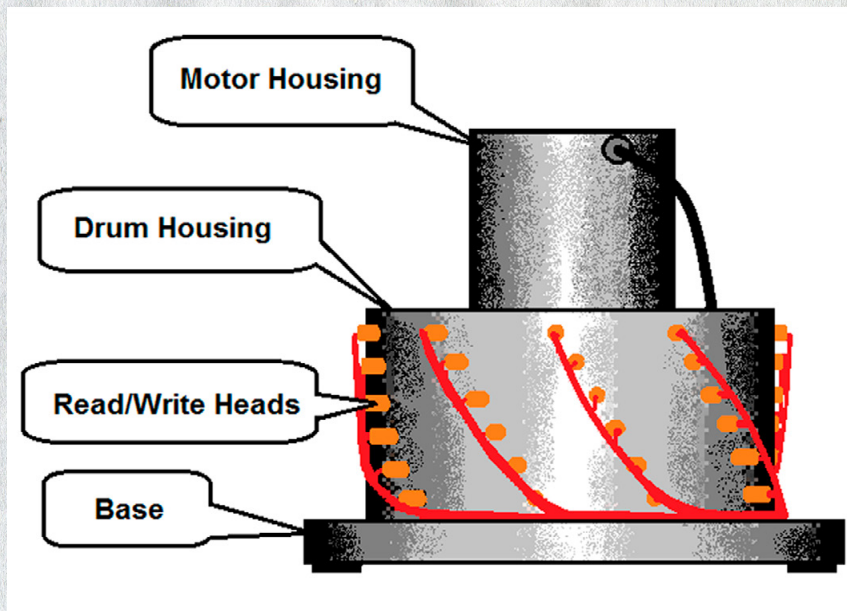
Kokeilin irrottaa toisen lukupään ja huomasin, että senkin raita oli pilalla. Irrotin lukupään toisensa jälkeen ja yritin etsiä mahdollisesti ehjiä raitoja. Löysin lopulta vain neljä ehjältä vaikuttavaa raitaa.

Kun rummun pintaa koskettaneet lukupäät säädettiin ulospäin, rumpu lähti pyörimään moottorin saadessa virtaa. Laitteen sisällä oleva suuri pyörivä massa teki siitä kuin jättimäisen gyroskoopin. Pyöriessään se pysyikin helposti pystyssä ilman tukea.

Päätin kokeilla tallentaa ja/tai toistaa pulsseja pyörivältä magneettipinnalta. Aiempi kokemukseni vastaavien laitteiden parissa liittyi nauhureiden äänipäihin, joten arvelin rummun luku- ja kirjoituspäiden käytön olevan samanlainen prosessi.

Omistin 5-tuumaisella näytöllä varustetun oskilloskoopin, joka oli tarkoitettu äänilaitteiden ja televisioiden kanssa työskentelyyn. Se oli tarkoitettu vaihtovirralla, joten ainoastaan muuttuvat signaalit näkyivät näytöllä. Sen taajuusvaste ylsi ehkä 200 kilohertsiin.

Tutkin magneettirumpua oskilloskoopilla ja yritin selvittää, toistuiko neljän ehjän raidan lukupäiltä mitään dataa. Kun säädin oskilloskoopin tulotason maksimiin, kaikkien signaali näytti hyvin samanlaiselta. Mukana oli pieniamplitudista vaihtelua, joka näytti kohinalta. Kohinassa kuitenkin toistui kuvio jokaisella rummun kierroksella, joten en voinut olla varma, oliko kyseessä vanha data vai ei. Päätin siis kokeilla tallentaa rummulle jotakin.



Piirros magneettirummusta, jossa näkyvät luku- ja kirjoituspäät.

Nauhurin äänipäät tallentavat signaalin magneettinauhalle 10–20 mA:n virralla, joten päätelin rummun kirjoituspään vaativan suunnilleen samansuuruisia virtaa. Kokosin piirin, joka ohjasi pulsseja kirjoituspäälle oikealta vaikuttavalla amplitudilla. Lyhyen pulssien kirjoituskokeen jälkeen tarkkailin rummun lukupään signaalia oskilloskoopilla määrittääkseni, olinko onnistunut muuttamaan kohinaa. Vaikutti siltä, että näin ei ollut käynyt. Oskilloskoopin rajoitusten vuoksi en voinut olla varma, epäonnistuiko kirjoittaminen vai enkö vain nähnyt, mitä olin kirjoittanut. En uskaltanut lisätä kirjoitusvirtaa kovinkaan paljon, jotta en kuumentaisi kirjoituspäätä ja tuhoaisi jälleen yhtä rummun raitaa. Laitoin koko magneettirummun sivuun ja aloin tutkia muita tapoja muistin rakentamiseen.

Kokeiltuani keksin tavan siirtää ääninauhan magneettipinnoitteen hartsista valetun sylinterin pinnalle. Vaikutti siltä, että tällä tavoin voisin valmistaa oman pienen rumpuni ja käyttää sitten ääninauhurin luku- ja kirjoituspäitä sen kanssa. Mekaanikon kykyne eivät kuitenkaan riittäneet tämän vision toteuttamiseen.

Kokeilin tehdä muistin käyttäen nopeasti pyörivää magneettinauhasilmuksia ja nauhurin äänipäitä. Tätä suunnitelmaa haittäsi eniten se, että käytössäni oli vain rajallisesti käsityökaluja. Jos olisin jatkanut tätä projektia, olisin ehkä saanut sen toimimaan, mutta lopputulos olisi ollut erittäin hidas.

Ostin tutuksi tulleesta Charles "Bud" Cokerin ylijäämämyymälästä erän pieniä kondensaattoreita, aikomuksenani valmistaa eräänlaista dynaamista muistia, joka luki ja virkisti jatkuvasti kuhunkin kondensaattoriin tallennettua varausta. Tämä oli osittain mekaaninen asetelma. Kokonaisuudessaan tähän olisi vaadittu 1 280 kondensaattoria ja huomattavasti parempi mekanismi kuin minkä sain rakennettua. Jälleen kerran nopeus olisi ollut heikko, jos järjestely siis olisi edes toiminut.

Tutkin akustista muistia ja yritin synnyttää tarvittavan viiveen eri tavoin. Kokeilin sähkökitaroiden kaiku-laitteissa käytettyä pitkää joustaa. Yritin synnyttää oikean mittaisen viiveen myös lähettämällä äänipulssit pitkän ilmatäyteen putken läpi.

Mietin, miten voisin käyttää oskilloskoopiani muistina Williams-putken periaatteen mukaisesti. Williams-putki tallentaa tiedot pieninä sähkövarauksina kuvaputken ruudulle. Tämä lähestymistapa jäi idea-asteelle.

Kokeilin asioita toisensa jälkeen, mutta mikään ei tuottanut haluamaani lopputulosta.

Muutto Ohioon

Lopulta nelivuotinen pestini ilmavoi- missa oli päättymässä. Olin päättänyt, etten jatka palveluksessa vaan palaan siviilielämään. Tähän mennessä olin työskennellyt tietokoneprojektini parissa lähes kolme vuotta. Ilmavoimista lähdön jälkeen aion palata vanhempieni luo Ohioon. Tästä muutosta

seurasi vakava ongelma, sillä koko tietokoneprojekti piti jotenkin kuljettaa Kaliforniasta takaisin Ohioon, lähes 4 000 kilometrin päähän.

Suuren piirilevyjä täynnä olevan kaapin, kirjoituskoneen ynnä muun lisäksi minulla oli laatikoitain osia, työkaluja, epäonnistuneita kokeiluja ja tarvikkeita. Otin yhteyttä kuljetusyritykseen selvittääkseni, mitä materiaalin lähetyksen kotiin maksaisi. Ilmeni, että kustannus olisi suurempi kuin mihin minulla olisi varaa, joten Ohioon lähettävän materiaalin määrää oli vähennettävä.

Aloin käydä tavaroita läpi päättäen jokaisen kohdalla, tarvitsenko sitä oikeasti. Tiesin, että pitäisin tietyt tavarat, kun taas toisten oli aika lähteä.

Raskaan magneettirummun kohdalla tunsin vahvaa halua heittää se menemään, koska voisin säästää paljon painoa, eikä laite tuntunut koskaan toimivan. Epätoivoisena kytkin sen vielä kerran päälle ja liitin oskilloskoopin toimivaan lukupäähän. Kun käänsin oskilloskoopin tulotason maksimiin, saatoin nähdä pienen, toistuvan kohinasignaalin. Päätin kokeilla kirjoittaa todella vahvan pulssin rummulle. Kytkin pikaisesti kirjoituspään johdot taskulampun paristoon. Tiesin, että lyhyen hetken ajan kirjoituspään läpi kulkisi monta kertaa vahvempi kirjoitusvirta kuin mitä olin ennen kokeillut. Kytkin sitten johdot takaisin oskilloskooppiin. Lukupäältä toistui nyt valtava pulssikuvioiden. Muutaman minuutin kuluttua olin onnistunut kirjoittamaan ja pyyhkimään lisää pulsseja.

Lähes romuna pitämäni magneettirumpu olikin hetkessä yksi arvokkaimmista omistamistani esineistä. Kehitystä rummun kanssa ei kuitenkaan ollut vielä aikaa jatkaa, sillä tavarat piti pakata Ohioon muuttamista varten.

Seuraavassa numerossa kerron, miten tietokoneprojektini tuli päätökseensä Ohiossa – ja ensimmäisistä ohjelmistani. 🎮

Tarina huipentuu Skrollissa 2019.4.

MVRMC, Dragon Multi, C7061

Philips VIDEOPACIN massamuistit

Nyky aika on saapunut myös vanhojen Philips-pelikoneiden harrastajien iloksi.

Teksti: Ismo Utriainen, Mikko Heinonen

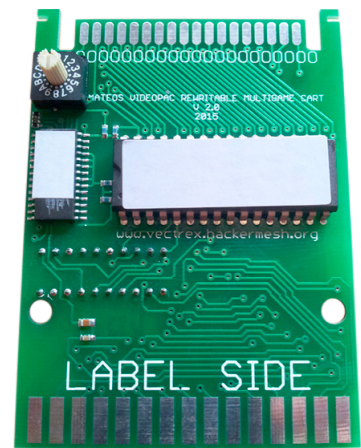
Kuvat: Mateos, Dragon, Wikimedia Commons (Studio Alijn), Ismo Utriainen, Janne Sirén

Philips Videopac -konsoleille (ks. Skrolli 2018.4) 70- ja 80-lukujen vaihteesta on olemassa kolmen tyyppisiä pelejä. Ensimmäkin on alkuperäisiä G7000-mallille tehtyjä pelejä, jotka toimivat kaikilla Videopac-konsoleilla. Lisäksi löytyy uudistettuja vanhoja pelejä, jotka toimivat G7000-konsolilla, mutta joihin uudemmalla G7400-konsolilla tulee monivärinen taustakuva. Nämä pelit on yleensä merkitty plus-merkillä pelin nimen perässä. Kolmatta ja harvinaisinta pelityyppiä edustavat vain G7400:lle tarkoitettut pelit, joita on olemassa ainakin kolme erilaista. Tarkkaa määrittelyä vaikeuttaa se, että osasta G7400-pelejä on kuitenkin olemassa myös ”plussattomat” versiot Odyssey2:lle, joka oli G7000:n amerikkalainen versio.

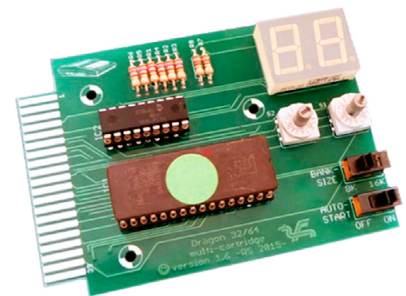
G7400-koneiden omistajille uustuotantomoduulit ovat käytännössä ainoa kohtuuhintainen keino päästä kokeilemaan niitä muutamaa peliä, jotka on tehty vain G7400-laitteelle. Koneen vaatimattoman menestyksen vuoksi myös pelien myyntimäärät olivat vähäisiä, joten myyntiin päätyvien kappaleiden hinnat ovat tiukasti etukenossa. Onneksi Videopacille löytyy uustuotantona ainakin kolmea erilaista moduulityyppistä massamuistitarkaisua, joista kahden sisältöä voi päivittää itse.

Kolme vaihtoehtoa

Nimihirviön *Mateos Videopac Rewritable Multigame Cartridge* (vectrex.hackermesh.org) taakse kätkeytyy noin parinkymppin hintainen, hieman kotikutoisempi monipelikasetti, joka



MVRMC



Dragon Multi

ei oletusarvoisesti sisällä yhtään peliä. Ostajan pitääkin toimittaa haluamansa pelit sen valmistajalle, jos haluaa moduulin valmiiksi ohjelmituna. Parinkymppin hintaan tarjotaan vielä lisälaitetta, jolla pelejä voi itse poistaa ja lisätä. Yhteensä piirille mahtuu 16 peliä, joista haluamansa voi valita mekaanisella kytkimellä pienen numeronäytön ja manuaalin avustuksella.

World of Dragon -sivuston (worldofdragon.org) kautta tarjotaan *Dragon Multi Cartridge* -nimistä vaihtoehtoa, jota voi käyttää adaptereiden avulla



C7061 ja G7400

useammassa vanhassa tietokoneessa ja konsolissa. Moduuli on alun perin tarkoitettu käytettäväksi brittiläisen Dragon-kotitietokoneen kanssa, mutta adapterien avulla se tukee Vectrex-, Odyssey2- ja Philips G7000/G7200/G7400 -konsoleita, Dragonille lähisukua olevaa Tandy Color Computer-mikroa sekä 1292 Advanced Programmable Video System -konsoleita. Ne olivat 1980-luvun alun halpalaitteita, joista Suomessa tunnetaan ainakin Emerson Arcadia, Hanimex ja Intervention 2001.

Moduuliin mahtuu 64 peliä, jotka voi itse valita. Moduulin sisällä pelit tallennetaan yhteen 27C4001 EPROMiin, jonka ohjelmointi onnistuu tavikseltakin, jos kotoa löytyy oikeat välineet. Tämänkin moduulin pelinvalinta tapahtuu mekaanisesti pienen numeronäytön ja manuaalin avulla. Dragon Multi Cartridgen ostaminen on hieman hankalampaa kuin kahden muun, mutta AtariAge -foorumia seuraamalla löytää tietoa varastotilanteesta ja hinnoittelusta. Yhdessä Videopac-lisäkortin kanssa hinta muodostunee suunnilleen samaksi kuin C7061:n, josta lisää alla.

Packrat Video Gamesin (packrat-vg.com) tarjoama C7061-moduuli erottuu ominaisuuksiltaan muista. Moduulissa on ohjelmallinen valikko, josta peli valitaan peliohjaimilla, sekä ehdottomasti suurin, yli 200 pelin ja ohjelman valikoima. Ainakaan

valmistajan tarjoamana tämän moduulin sisältöön ei voi itse vaikuttaa, mutta sen tarjonta käsittää lähes kaikki Videopacille julkaistut pelit, joukon skenepelejä ja kattavan kokoelman testiohjelmaa. Moduuli on hyvin ammattimaisen oloinen ja muodoiltaan tarkka



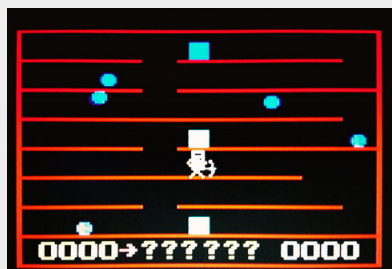
Videopac G7400:n pelejä

Trans American Rallyssa ohjataan takaapäin kuvattua autoa, jota peruuttavat kamikaze-autot ahdistelevat. Tältä ainakin tuntuu, koska osa vastaan tulevista autoista muuttavat kurssiaan kohti pelaajan autoa, kun niitä yrittää väistää. Pelin ohjaus on melko kamala, eikä sen pelaaminen tunnu hauskalta edes vuoden 1984 mittapuulla.

Helicopter Rescue on mielenkiintoinen tankkiammuskelu, jossa ohjataan helikopteria kentällä, joka liikkuu sekä horisontaalisesti että vertikaalisesti (ei tosin samaan aikaan). Pelin ohjaustuntuma on

sujuva, ja aikanaan sitä olisi varmasti esiteltu kavereille. Tätä ei kannata sekoittaa samannimiseen peliin, joka julkaistiin G7000:n jenkki-versio Odyssey2:lle.

Norseman on strategiapeli, jossa ohjataan hahmoa heksaruuduilla. Tämän enempää kirjoittaja ei siitä sitten ymmärtänytkään. Packrat Video Gamesin mukaan kyseinen peli ei ole vain G7400:lle tarkoitettu, vaan plus-peli, joka on värikkäämpi G7400-konsolilla kuin G7000:lla pelattaessa – toinen tällainen plus-esimerkki on *Pickaxe Pete* (kuvassa G7000- ja G7400-näkymät).



Commodore 64 ja MIDI

SID
soikoon

Valjasta Commodore 64:n
uniikki sointi omaan
orkesteriisi.



Teksti ja kuvat: Jarkko Lehti

Moni tuntee 1980-luvun kotitietokonekuninkaan Commodore 64:n legendaarisen SID-äänipiirin. Joku tietää myös, että on olemassa SID-piirin ympärille kehitettyjä syntetisaattoreita, kuten *Elektron SidStation*, jota myytiin 2000-luvulle asti, tai tuorempi *sammichSID*.

Jostain syystä allekirjoittaneen kokemuksen mukaan juuri kukaan ei tunnu kuitenkaan kuulleen siitä, että myös ihan sitä tavallista Commodore 64 -tietokonetta voi sopivalla ohjelmistolla ja MIDI-liitännällä käyttää syntetisaattorina.

Pienen budjetin musisoijalle Commodore 64 onkin hieno lisä soitinarsenaaliin, joskin vaadittava MIDI-liitäntä on perinteisesti ollut hieman kimurantti hankittava.

Aloitetaan liitännästä

C64:n MIDI on toteutettu yleensä kytkemällä liitännän tiedonsiirto NMI-keskeytykseen (*Non-Maskable Interrupt*), jota ei nimensä mukaisesti voi estää. Näin saadaan aikaiseksi hyvin vakaa kellotus, oli suoritin miten kiireinen hyvänsä suorittaessaan muita tehtäviä. Teknisesti C64 soveltuukin erittäin hyvin syntetisaattoriksi – kunhan löytää liitännän.

Vajaalla 50 eurolla on saatavissa *Mssiah*-niminen MIDI-moduuli (ms-siah.com), mutta se on lukittu omalle ohjelmistolleen, ja liitäntää ei näin ollen voi hyödyntää muiden ohjelmien kanssa. Vaihtoehtona aikanaan myyty *DATEL*-MIDI-sovitin on yhteensopiva

useimpien ohjelmistojen kanssa, mutta niiden saatavuus on hieman kehoa sovitin harvinaisuuden vuoksi.

Ongelman ratkaistakseen saksalainen **Frank Buss** kehitti kaikkien MIDI-moduulien äidin: *Kerberos C64/C128 MIDI Flash Interface* -nimisen moduulin. Nimi *Kerberos* tulee kolmesta MIDI-portista moduulin päällä, viitaten myyttisen Kerberoksen kolmeen päähän. Moduuli on avointa koodia niin softaa kuin rautaakin myöten, joten osaava rakentaa laitteen itse – vähemmän näppärien on ostettava se jostain.

Kerberos on ollut pitkään loppuunmyyty, mutta laitteen tekijä on luvannut uutta erää saataville vielä tämän vuoden puolella *Protovision*-verkkokauppaan (protovision.games).

Syntetisaattoriohjelmit

Käsitlemme tässä Commodore 64:n MIDIä nimenomaan *Kerberos*-liitännän (ja myös *DATEL*in) näkökulmasta. Musisoimaan pääsee kytkemällä MIDI-soittimen, esimerkiksi kosketinsoittimen, kiinni sovitin MIDI-IN-liitäntään ja lataamalla C64:lle syntetisaattoriohjelman. Mutta kun liitäntä on tehty, mikä ohjelmistoksi?

Kuranteja Commodore 64 -syntetisaattoriohjelmistoja ei ole liikaa, mutta pari mainitsemisen arvoista vaihtoehtoa löytyy: *Cynthcart* (qotile.net/cynthcart.html) ja *Station64* (djindikator.net/#c64) tarjoavat molemmat monipuoliset mahdollisuudet omien äänien kustomointiin ja rakentamiseen. Mukana tulevat oletussoundit ovat myös riittävän hyviä alkuun pääsemiseen.

Vakavampi lähestyminen vaatii sitten jo omien soittimien luomista tai valmiiden soittimien kustomointia vastaamaan oman SID-yksilön sointia. SID-äänipiiri on nimittäin varsin uniikki tapaus, johon liittyy valtavasti myyttejä ja sekaannusta.

Yksilöllinen SID

MOS 6581 tai tuttavallisemmin SID (*Sound Interface Device*) on **Robert Yannesin** Commodorelle suunnittelema äänipiiri. Piiriä käytettiin Commodore 64:ssä (1982) sekä myöhemmin Commodore 128:ssä (1985). Commodorella oli suunnitelmia käyttää kahta SIDiä stereona myös julkaisemattomassa Commodore 65:ssä (noin 1991). Juuri alkuperäinen kohde, Commodore 64, teki kuitenkin piiristä legendan.

Historia kertoo, että Commodore 64 kasattiin aikanaan kovalla kiireellä, eikä piirien luomiselle jäänyt riittävästi aikaa, joten niihin jäi ”ominaisuuksia”, jotka edelleen tänä päivänä aiheuttavat kiistelyä. SIDin ominaisimman soundin tuottaa *filteriksi* kutsuttu suodin, jolla on kolme eri tilaa: ali-, kaista- ja ylipäästö. Tämä on piirin temperamenttisuuden ydin.

Digitaaliseen piiriin piti saada analoginen suodin, mutta sen toteuttaminen halvoilla puolijohteilla oli haastavaa Commodoren omistamalle MOS-puolijohdtehtaalteille aikakauden teknologian vuoksi. Digitaalitekniikka on helppoa – sähkö joko kulkee tai ei kulje – mutta analogipuolella onkin merkitystä, kuinka paljon sitä sähköä kulkee. Näin ollen jopa saman tuotantoerän SIDit kuulostavat erilaisilta.

Väittäisinkin, ettei maailmassa ole kahta soinniltaan täysin samanlaista piiriä. SID-piirien äänen erilaisuus tuokin aivan uuden ulottuvuuden syntetisaattorikäytössä, sillä piiri, joka toistaa peli- ja skenemusiikkia ”väärin” voikin olla hyvin käyttökelpoinen villien äänien luomisessa. Ei siis kannata heittää huonoja piirejä roskeen tai antaa pois kokeilematta niitä ensin syntetisaattorikäytössä.

Soinnin tekniikkaa

SIDin ääni perustuu niin kutsuttuun ADSR-malliin (*Attack Decay Sustain Release*), jossa äänelle määritellään aloitus-, päästö-, pito- ja vapautusai-ka. Näillä perusparametreilla voidaan määrittellä äänen luonnetta. Lisäksi SID-piirissä on valittavissa neljä aaltomuotoa: pulssi, kantti, saha ja kohina, sekä jossain määrin näiden yhdistelmiä, joiden toimivuus on myös piiri-kohtaista.

Lisäksi oskillaattoreilla on mahdollista tehdä ring-modulaatio, jossa yhdellä oskillaattorilla ohjataan toista. Syntetisaattorikäytössä on myös ohjelmistosta riippuen mahdollista käyttää useampaa SID-piiriä, jotta äänelle saadaan lisää kanavia käyttöön. SIDissä on kolme äänikanavaa, ja kahdella piirillä määrä tuplataan kätevästi kuuteen, jolloin äänen variaatioon on enemmän vaihtoehtoja.

SID itsessään on vain tyhmä laite, jota pitää aktiivisesti ohjata ohjelmallisesti, jotta ääneen saadaan eloa. Peli- ja demokäytössä tavanomainen SID-virkistystaajuus on 50 hertsiä, mutta nostamalla päivitystaajuudeksi esimerkiksi 200 tai jopa 400 hertsiä saadaan aikaiseksi paljon villimpiä ääniä, joita onkin päästy vuosien varrella ihmettelemään demoskenen tuotoksissa.

Mitä enemmän suoritinaikaa käytetään SIDin ohjaamiseen, sitä vähemmän sitä jää muille tehosteille. Siksi näitä niin kutsuttuja *multispeed*-kappaleita ei juurikaan kuule demoissa, saati peleissä. Syntetisaattorikäyttöä tämä ei kuitenkaan juuri rajoita.

6581–8580–FPGA

Alun perin Commodore 64 varustettiin tyyppinumeron 6581 SID-äänipiirillä. Näissä piireissä on todella paljon eroja eri yksilöiden välillä. Kun Commodore alkoi myöhemmin rakentaa *cost reduced*-mallia Commodore 64:stä, myös piirit suunniteltiin uusiksi uudemmalla piiriprosessilla. Näin saatiin paremmin hallintaan eri yksilöiden variaatiot. Uudempi 8580-piiri soi yhtenäisemmin.

Vaikka piirit äkkiä kuunneltuna ovatkin samanlai-

set, ne ovat hyvin erilaiset juuri filteerin osalta, joka on mielestäni C64:n soundin sielu ja ydin. Kuuluisasti 8580 korjasi esimerkiksi erään 6581-piirin ”bugin”, jota kekseliäät ohjelmoijat olivat ryhtyneet hyödyntämään yksinkertaisten äänisamplejen toistamiseen – 8580-piirillä nämä kikkaänet jäivät vaimeiksi. Eroja on kuitenkin paljon muitakin.

Nykyään markkinoilla on myös erilaisia uudelleenluotuja SID-korvikkeita. Osa perustuu äänen emulointiin ohjelmallisesti ja uudemmat nojaavat FPGA-teknologiaan, jossa SID-rauta on luotu uudelleen käyttäen uudelleenohjelmoitavaa piiriteknologiaa. Kannattaa tutustua esimerkiksi osoitteeseen fpgasid.de sekä Gideon's Logicin Ultimate -tuotteisiin (ultimate64.com).

Mikä piireistä kuulostaa parhaalta, onkin sitten syntetisaattorikäytössä aika toissijainen kysymys, sillä viimeinen sana on artistin itsensä käsissä. 🎸

Katso video Skrollin YouTubesta: youtube.com/c/skrollimagazine



Iso- ja pikkuveli: Punainen Kerberos-MIDI-liitäntämoduuli poseeraa yhdessä erään SID-syntetisaattorin, therapsidin kanssa.

ATOSTEK REKRY

IRTONUMERO 9,90 (SIS. LVV.) • KOKENUT MIKROILIJA, HAE NYT!



**MAHTUUKO HAKEMUS
LERPULLE?**

**KOVAN LUOKAN
TOIMISTOT VERTAILUSSA:
TAMPERE & ESPOO**

**HAUSSA MM. JÄRJESTELMÄYLLÄPITÄJÄ, ATK-KONSULTTI,
SOVELLUTUSOHJELMOIJA... HAE NYT!**

ATOSTEK.COM/REKRY84

ATOSTEK

Tramiel's Exit Device

Commodore 264

Tiedät kuusnelosen, viissatasten ja varmasti satakakskasinkin, mutta muistatko Commodore 264:n?

Teksti: Ismo Utriainen, Janne Sirén
Kuvat: Wikimedia Commons
(Babylon4, Cbmeeks, Toni Birrer, Knurrikowski), Ismo Utriainen, Janne Sirén

Kotitietokoneyhtiö Commodoren seikkailut 8- ja 16-bitististen tietokoneiden parissa 1980-luvulla ovat tuttuja suosituimpien mallien osalta, mutta Commodore kilpaili myös omien suosikkiensa kanssa tuomalla markkinoille niin kutsutun *Commodore 264* -sarjan.

Sarjan isä oli yhtiön perustaja ja toimitusjohtaja **Jack Tramiel**, joka esitteli CES 1984 -messuilla neljä 8-bittistä tietokonetta: Commodore 116, 232, 264 ja V364. Osassa olisi sisäänrakennettu hyötyohjelmisto, jonka käyttäjä saisi ostotilanteessa valita, ja sisälle asennettaisiin sopivat ROMit. Toimisto-ohjelmisto sisältäisi jopa ohjelmien rinnakkaisajon. V364:ssä olisi lisäksi numeronäppäimistö ja puhetoiminto¹.

Kolme päivää myöhemmin Tramiel lähti Commodorelta.

Monitaituri TED

Commodore 264:stä tuli yleisnimi tuoteperheelle, joka rakennettiin Commodoren omistaman MOS Technology -piiri-yhtiön valmistaman *TED*-piirin ympärille (*7360/8360 Text Editing Device*). Nimensä mukaisesti TED suunniteltiin vastaamaan kuvapuolesta. TED ei kuitenkaan ollut pelkkä videoopiiri, vaan siihen integroitiin myös äänet, ajastimet sekä näppäimistön ja muistin hallintaa.

TED ja Commodore 264 -sarja kehitettiin, koska Commodore 64 oli liian kallis valmistaa – siitä ei olisi alle sadan dollarin hintaluokkaan. Historiankirjoitus mainitsee useammankin potentiaalisen kilpailijan: Haluttiin haastaja Sinclairin halvalla ZX Spectrumille tai Mattel Aquariuksen (ks. Skrolli 2018.3) kaltaisille amerikkalais-



sille halpatuotoksille. Toisaalta pelättiin japanilaisia.

Olivat syyt mitä tahansa, kiistatonta on, että siinä missä C64 käytti erillisiä video-, ääni- ja tukipiirejä, Commodore 264 -perheessä monet toiminnoista oli integroitu TED-sirulle. Näin ollen koneissa oli vähemmän komponentteja. TED tarjosi myös enemmän värisävyjä (121) kuin C64:n VIC-II-piiri (16), mutta se ei tukenut spritejä. Äänissä TED häviää selvästi C64:n legendaariselle SIDille.

Sarja ei kuitenkaan päässyt hintavoittoiteisiinsa, ja Tramielin lähtö suisti suunnitelmat muutenkin raiteiltaan. Lopulta julkaistiin vain Commodore 116 sekä Commodore 264 -mallia vastannut Commodore Plus/4, joiden ulkonäössä näkyy 264-perheen design. Muutamia esituotantoversioita julkaisemattomista malleista, kuten alkuperäisestä 264:stä, kiertää keräilijöillä.

Itse asiassa sarjan tunnetuimmaksi tuotteeksi nousi aivan toisen näköinen kone: vanhan Commodore 64:n ulkomuodon saanut Commodore 116 -variantti, Commodore 16.

Ekosysteemin tappaja

1980-luvulla ymmärrys tietokone-markkinoista ei ollut nykypäivän tasolla. Koneita suunniteltaessa lisälaitteiden, ohjelmien ja pelien luoman ekosysteemin merkitystä ei täysin tiedostettu. Niinpä Commodore 264 -tuoteperhe ei ollut yhteensopiva suo-

situn Commodore 64 -mallin kanssa – virhe, jonka yhtiö korjaisi myöhemässä Commodore 128:ssa.

Myönnytyksenä ekosysteemiajattelulle 264-tuoteperhe oli kuitenkin keskenään yhteensopiva. Commodore 16 olikin 264-perheen merkityksellinen tuote. Ei tosin vain siksi, että se myi eniten, vaan siksi, että se asetti rajat koko tuoteperheen ohjelmistokehitykselle. Kuten nimestä voi päätellä, Commodore 16 sisälsi alkuperäisen ZX Spectrumin tapaan kuusitoista kilotavua muistia.

16 kilotavua oli kohtuullisesti Spectrumin julkaisuvuonna 1982, mutta Commodore 16:n ilmestyessä vuonna 1984 se oli jo vähän. Ohjelmille muistia jäi vieläkin vähemmän, sillä ajan tapaan osaa siitä käytettiin muuhunkin. Kun vielä Commodore 116:ssakin oli sama muistimäärä, 16 kilotavun RAM-muisti sinetöityi 264-perheen perustasoksi.

Commodore 264 -sarjan suunnittelussa tehtiin muitakin kyseenalaisia valintoja ekosysteemin kannalta, esimerkiksi vaihdettiin kasettiaseman ja peliohjainten liittimet. Roppakaupalla kolmansien osapuolien tuotteita sekä Commodoren omia, muille tietokoneille tarkoitetuista lisälaitteista muuttui epäyhteensopiviksi.

Ajatuksena varmaankin oli, että Commodore myisi 264-koneisiin omia peliohjaimiaan tai että koneen vaihtaja ostaisi myös uuden kasettiaseman. Liitännät olivat kuitenkin elektronisesti yhteensop-

¹ Commodore V364:n sisäänrakennettu puhetoiminto Toshiba T6721A -äänipiireineen julkaistiin C64:lle Magic Voice -moduulina.

pivia, joten useimmat ratkaisivat ongelman sovittimilla, varsinkin peliohjainten osalta. Commodoren kalliit levyasemat sentään toimivat ristiin, joskin 264-perheelle oli myös oma 1551- asemansa.

Plus/4:stä Minus/60

Siinä missä Commodore 16 oli ulkoisesti kotelon väriä lukuunottamatta kopio C64:stä ja VIC-20:stä, Commodore 116 oli selvästi pienempi. Siinä oli myös tuon ajan halpakoneista tuttu kuminäppäimistö. Teknisesti laite on samankaltainen kuin Commodore 16 ja hintakin oli samaa tasoa, joten ostajat päätyivät sen sijaan usein C16:een. 116-mallia myytiinkin vain Euroopassa.

Yrityskäyttöön suunnattu Plus/4 muistutti ulkoisesti Commodore 116:ta, mutta se oli varustettu paremmalla näppäimistöllä. Lopullinen nimi ei tällä kertaa viitannut muistimäärään lainkaan, sillä Commodore Plus/4:ssä oli 64 kilotavua muistia. Plus/4:n nimi tuli siitä, että koneen kiinteässä ROM-muistissa oli neljä hyötyohjelmaa.

Mukana tulivat tekstinkäsittely-, taulukko-, tietokanta- ja grafiikkaohjelmat, joskin hurjimmista CES-suunnitelmista oli luo-

vuttu. Ohjelmapaketti oli yksinkertainen ja kiinteä. Kaikissa Commodore 264 -sarjan koneissa oli myös sisäänrakennettu konekielimonitori, TED-MON.

Vaikka Plus/4 ei jäänyt täysin historian kuriositeetiksi, Commodore 16:n rajoittava vaikutus sen ohjelmistovalikoimaan on kiistaton. Commodore 16 epäilemättä lisäsi Plus/4:n ohjelmisto- ja etenkin pelitarjontaa, mutta liian harva ohjelmistokehittäjä uskalsi hyödyntää Plus/4:n laajempaa 64 kilotavun keskusmuistia.

Commodore 16:lle oli saatavilla muistinlaajennus, mutta ZX Spectrumin tapaista² hittiä siitä ei tullut. Ti-

² Toisin kuin 8-bittiset Commodoret, ZX Spectrum -perhe oli pitkälti ohjelmistoyhteensopiva 16 kilotavun mallistaan laajennettuun 48 kilotavuun sekä myöhempiin 128 kilotavun malleihin asti.

lanne olisi saattanut olla toisenlainen, ellei Commodore 16:n lisämuistin pahin kilpailija olisi löytynyt niin läheltä: itse Commodore 64. Siinä missä C64 pysyi myynnissä hurjat 12 vuotta, Länsi-Euroopassa 264-tuoteperehettä myytiin pääasiassa parin vuoden ajan 1980-luvun puolivälissä.

264-sarjan koneita myytiin yleisten arvioiden mukaan yhteensä noin miljoona kappaletta. Viimeinen hurraa sarjalle tuli Spectrumin tapaan kommunistisessa itäblokkissa, johon 264-sarja päätyi Unkarin hieman avoimemman markkinan kautta, ja Meksikossa. Vertailun vuoksi ZX Spectrumia myytiin viisi miljoonaa ja C64:ää 13–17 miljoonaa kappaletta. 🐛

Lue myös: *Ei näin!* – Commodorin kääntöpuoli, Skrolli 2014.4. Vanhojen Skrollien pdf-versiot: skrolli.fi/numerot



Commodore	16	116	232	264	Plus/4	V364
Julkaisu	1984	1984	-	-	1984	-
Piirit	TED	TED	TED	TED	TED	TED + T6721A
RAM (kt)	16	16	32	64	64	64
ROM (kt)	32	32	32–64	32–64	64	64–80 (96)
ROM-ohjelmat	-	-	Optio	Optio	Kiinteä 3+1	Optio + Magic Voice

Kakskuusneloet nykyaikaan

Jos hyllystä sattuu löytymään 264-sarjalainen – Commodore 16, 116, Plus/4 saati esituotantomalli – ole varovainen virtälähteen kanssa: C16:n ja 116:n virtälähde on korvattavissa millä tahansa yhdeksän voltin virtälähteellä, mutta virtaliittimessä plus on rungossa ja miinus navassa. Plus/4:n virtaliitin taas on vastaava kuin Amigassa ja C128:ssa, mutta sisuksiltaan virtälähde onkin C64:n.

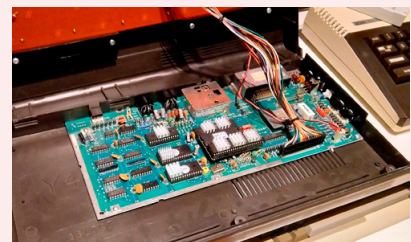
Kannattaa myös avata kone ja liimata jäähdityssiilit erityisesti TED-piirin päälle. TED kuumenee ja sillä on maine hajoilevana piirinä, jolle ei löydy korvaajaa nykytuotannosta. Prosessori on helpompi: MOS 7501/8501:n voi korvata 6510-suorittimella, joka löytyy

muun muassa Commodore 64:stä – tarvitaan vain pieni sovitin ja EPROM, jossa on yhteensopivuuden varmistava koodi.

Commodore 16:n muistinlaajennukseen löytyy nykytuotannosta useita ratkaisuja, joista helpoin asennettava on prosessorin alle kantaan sijoitettava lisämuisti. Astetta vaikeampi tee-se-itse-projekti on muutaman muistipiirin vaihto, parin radan katkaisu ja jokusen hyppylangan kolvaus paikalleen. Moduuliporttiin laitettavat lisämuistit eivät täysin vastaa koneen sisäistä muistia, joten niiden kanssa kaikki pelit ja ohjelmat eivät välttämättä toimi.

Commodore 64 -koneelle tarkoitettut SD2IEC-levyasemaemulaattorit (ks. Skrolli 2018.2) toimivat myös Commodore 264 -sarjassa. Verkkokaupois-

ta (esim. amigastore.eu) löytyy myös sovitin, jolla koneisiin saa kiinni yleisemmän Atari-liittimellä varustetun peliohjaimen – vaikkapa uustuotetun Competition Pron (ks. Skrolli 2018.1). Lisätietoja: plus4world.powweb.com/hardware.



Commodore 16, johon on lisätty muistia ja jäähdityssiilejä. TED on iso piiri kuvan keskellä, kolmen lisätyn jäähdityssiilin alla. C64-pohjainen kotelo jää todella väljäksi.

Commodore/Spectrum -sisäpiiri 2019

Janne Sirén

Tätä kirjoittaessani takana on vuosi *Sirénin sisäpiiri* -palstaa Skrollissa ja käsillä on jo viides sisäpiiri. Vuosi sitten aloitin itse-oikeutetulla Commodore-sisäpiirillä (Skrolli 2018.3). Sen jälkeen vuorossa oli Sinclair Spectrum (2018.4), joka olisi kenties ansainnut oman ”Spectrum-sisäpiirinsä” jo 1980-luvun tietokonelehdissä – ellei maahantuonti Suomeen olisi päättynyt kriittisellä hetkellä Sinclair-yhtiön talousvaikeuksiin. Virtuaaliodellisuus- (2019.1) ja robottiautosisäpiirit (2019.2) olivat sitten loikkia aivan toiseen – modernimpaan – suuntaan.

Modernejakin sisäpiirejä tehdesä huomasi, ettei historia ole koskaan kaukana. Id Softwaren **John Carmack** on Oculus Riftin takana, mutta moniko muistaa, että saman miehen DOOMia ja Quakea pelattiin virtuaalilaseilla ensikertaa jo 1990-luvulla. Nykypäivän ”uusi” keinotodellisuus on paljosta velkaa vanhoille virtuaalilyöpärille. Sinclair C5 -sähköauto (1985) taas ei mullistanut maailmaa, mutta teki kuitenkin tutuksi ajatukset sähköautosta kuluttajaelektronikkana ja autofirman johtajasta nörttien kultisankarina. Kuulostaako tutulta?

Kiitos, että olette jaksaneet mukana ensimmäisen vuoden. Kiitos myös Skrollin toiselle hovitahtajalle ja Suomen retroerikoisuuksien keräilijäkninkaalle **Manu Pärssiselle**, joka on väsymättä jaksanut kamppailla yksinkertaisen sisäpiiriestetiikan parissa. Kenties yllättävää kyllä, visuaalisesti koristeellisempia artikkeleita on usein helpompi taittaa, kuin tällaista yksinkertaista palstaa.

Kuten Applen hiljattain jättänyt muotoilija **Jony Ive** on todennut: ”todellinen yksinkertaisuus on paljon enemmän kuin sekamelskan ja koristelun puuttamista.” Onneksi Manu on Skrollin oma Jony, jonka ansiosta saat selvää tästäkin artikkelista. Myös sisäpiirin sisältö tähtää monimutkaisuuden järjestämiseen: tällä kertaa tarjolla on valmiiksi pureskeltu yhteenveto parin toisiinsa nivoutuneen teknologia-alakulttuurin viimeaikaisista tapahtumista.

Elämä on valintoja

Yhdeksi **Risto Siilasmaan** luotsaaman alkuperäisen Mikrobitti-lehden Commodore-sisäpiirin (1986–1988) ominaisuudeksi muodostui teemojen seuraaminen useamman numeron

Sirénin sisäpiiri -palstalla Janne Sirén ruotii seuraamiaan teknologiaailmiötä ja -alakulttuureita. Jatkamme nyt siitä, mihin Sirénin viime vuotiset Commodore- ja Spectrum-sisäpiirit jäivät.

ajan. Eräs tällainen teema oli Commodore 64:n GEOS-käyttöliittymä. Risto tosin vihjasi Mikrobittissä 4/1988, että pestinsä loppua kohden ainainen GEOSista puhuminen johtui enemmänkin ideoiden puutteesta kuin taroituksellisuudesta. Riston pääaihe, Commodore 64, kun oli vuoteen 1988 mennessä jo menettänyt uutisarvoaan 16-bittisille kotimikroille.

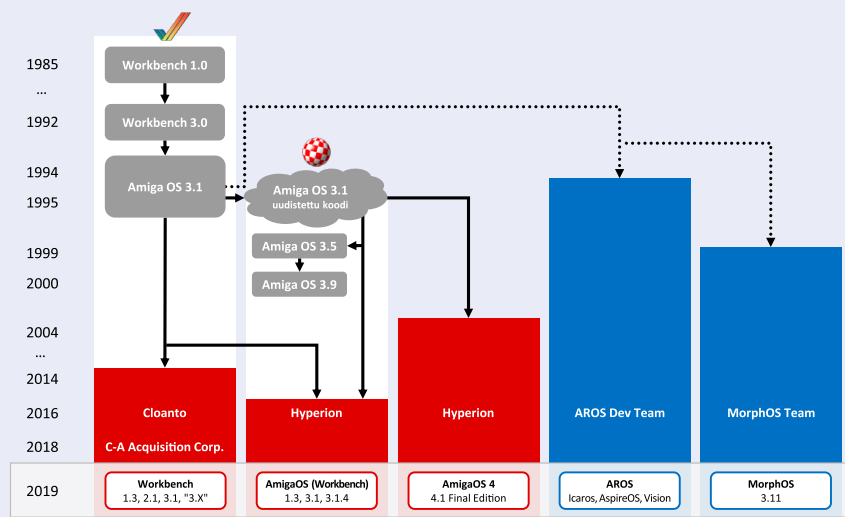
16-bittisten kotitietokoneiden kulta-ajan Risto itse jätti väliin, sillä hän ”riipusti Kuusnelosensa naulaan” ja siirtyi naputtelemaan tehokasta AT-yhteensopivaansa, kuten seuraava palstanpittäjä ja klassisen *Uno Turhapuro* -pelin tekijä **Pasi Hytönen** asian aikanaan muotoili. AT:lla viitattiin IBM PC/AT:hen ja laajemmin kaikkiin Intel 80286 -pohjaisiin, toisen sukupolven PC-yhteensopiviin tietokoneisiin – varhaiseen MS-DOS- ja Windows-rautaan siis.

Kun katsoo mihin Risto on työelämässään ehtinyt, ei voi välttyä ajatukselta, että tehokkaan AT-yhteensopivan naputtaminen sittemmin hyvinkin kuolleiden ja kuopattujen 16-bittisten kotimikrolegendojen sijaan oli viisas veto. Toinen vanha mikrobittiläinen – allekirjoittanut – ei siirtynyt PC-rau-

Amiga – kaikkien kaveri

Retrokoneilla on usein vähän vaikeaa. Mutta millään ei ole niin vaikeaa, kuin Amigalla.

Edellisen Commodore-sisäpiirin (Skrolli 2018.3) lisäksi kirjoitin Amigan käyttöjärjestelmien haajaannuksesta Skrollissa 2017.3. Amigan tilanne pääsi myös **Mikko Heinosen** Ei näin! -palstalle viime numerossa (2019.2). Jos tämä Commodore 64:n 16-bittisen





taan ennen kuin oli pakko (DOOM) ja naputti Amigaa 1990-luvun lopulle saakka. Vertailun vuoksi voidaan todeta, että suurten pörssiyhtiöiden hallitusten sijaan minun ”sisäpiirini” on 30 vuotta myöhemminkin edelleen vain tässä paperilla. Kiitos, Amiga.

Hyvänä puolena minun kohdallani Rahoitustarkastus ei närkästy, kun keron teille kaiken mitä tiedän. Menneistä valinnoista oppimisen ja eteenpäin menemisen sijaan hetki tuntuukin siis otolliselta seuraavalle 8/16-bittiselle Sirénin sisäpiirille. Ei kuitenkaan ideoiden puutteen vuoksi, vaan siksi, että kuluneen vuoden aikana aika moni muukin on valinnut – tai ainakin yrittänyt – jatkaa siitä, mihin viime vuoden Commodore- ja Spectrum-sisäpiireissäni jäätiin.

Johonkin minä kuitenkin vedän rajan: en puhu GEOSista. GEOSin sijaan aloitamme siitä naputtelusta,

joskin tehokkaan AT-yhteensopivan sijaan naputtelemme kasibittisiä. Tai ainakin yritämme. Näppäimistö on nimittäin osoittautunut 2010-luvun retroraudan suurimmaksi kompastuskiveksi.

Joko hatuttaa?

Rakensin Skrollissa 2018.1 ”Uusnelosen” eli uuden Commodore 64C:n Pixelwizard Retro Shopista ostetuilla uusiotuotetuilla kuorilla (shop.pixelwizard.eu) ja Individual Computersin C64 Reloaded MK2 -emolevyllä (icompe.de). Lähes kaiken olennaisen uustuotettua Competition Pro (Star)-peliohjainta myöten saa nykyisin uutena. Emolevyn kohdalla on jopa vaihtoehtoja, sillä Gideon’s Logicin Ultimate-64 (ultimate64.com) on sittemmin myös päässyt laajaan jakeluun – sen kanssa ei tarvitse välttämättä edes niitä muutamaa käytettyä mikropiiriä.

Uusnelonen. Tässä Individual Computersin ja Pixelwizard Retro Shopin uusio-osista rakennetussa Commodore 64C:ssä vanhaa on vain näppäimistö ja pari mikropiiriä (Skrolli 2018.1).

Lähes kaiken, paitsi näppäimistön. ”Uusneloseni” ainoa näkyvä käytetty osa oli näppäimistö, joka tuli vuoden 1986 Commodore 64C:stä (ks. kuva). Elätelin tuolloin tosin vielä toiveita, että näppäinhatut saisi sentään uusitua. Kerroin artikkelissa itävaltalaisesta nimimerkistä **BukoCharly** – taustalla veijarit **Michael**, Peter ja Tom – jotka keräsivät uudelleenheräteltyyn Phase 5 -yritysnimen (phase5.eu) turvin joukkorahoituksella lähes 45 000 euroa Commodore 64:n näppäinhatujen uuteen tuotantoon. Sittemmin vain Michael on jäänyt pakertamaan vuonna 2015 alkaneeseen projektiin.

Alkuvuoteen 2018 mennessä hän oli saanut aikaiseksi erän ruiskuvalettuja

seuraajan pidempi tarina kiinnostaa, suosittelen näitä kertauksena. Vanhat Skrollit ovat saatavilla pdf-muodossa osoitteesta skrolli.fi/numerot.

Toisin kuin Commodore 64, jonka harrastustoiminta on pitkälti klassisen retroalustan lämmittelyä vanhoille klassikoille ja satunnaisille uusille peleille, Amiga-teknologiaa on kaiken aikaa aktiivisesti suunnattu myös uudelle, kyvykkäämmälle ja tehokkaammalle raudalle. Toki Amigankin yhteydessä harrastetaan ”klassikkomalleja” – kuten hakataan vanhalla Commodore Amiga 500:lla tai 1200:lla Stunt Car Raceria sunnuntai-iltana tai Oulun

Taiteiden Yössä (Skrolli 2016.3) – mutta ei siitä sen enempää tässä. Mitä kuuluu Amigan kehitystyölle?

Edelleen kehittyvät Amiga-alustat ovat jakautuneet karkeasti viiteen leiriin, joista kolme moderneinta ovat belgialaisen Hyperion-yhtiön edistämä AmigaOS 4 (amigaos.net) sekä AmigaOS-yhteensopivat uudelleentoetukset – olen joskus kutsunut näitä vertauskuvallisesti ”distroiksi” – MorphOS (morphos-team.net) ja AROS (www.aros.org, aros.sourceforge.net) AROS-leirin sisällä on julkaistu myös useampia omia distroja, kuten Icaros ja AspireOS.

Lisäksi kaksi leiriä perustuvat AmigaOS:n vanhimpiin versioihin. Commodore- ja Amiga-oikeuksia haahtanut sekä emulaattoripaketteja kehittänyt Cloanto on kunnostautunut retropainotteisemmilla tuotteillaan, jotka perustuvat ostettuihin Commodore/Amiga-oikeuksiin, sekä osittain avoimen lähdekoodin tuotoksiin. Osin päällekkäin Cloanton kanssa myös Hyperion on julkaissut vanhempiin AmigaOS-versioihin perustuvia tuotteita, irrallaan AmigaOS 4:stä.

Avoimen lähdekoodin AROS-projekti etenee omaan verkkaiseen tahtiinsa ja MorphOS-käyttäjies-

näppäinhattuja ja vieläkin suuremman erän meriselityksiä. Viimeisimpänä tietona Michael kamppaili tuolloin näppäinmerkintöjen kanssa, joita hän oli saanut tulostettua näppäinhatuille vain joitakin ongelmallisia pieneriä. Valitettavasti tilanne ei ole paljoa parempi puolitoista vuotta myöhemmin elokuussa 2019, kun kirjoitan tätä palstaa.

Viimeisin sana suoraan Michaelilta on huhtikuulta. Tarkoitus oli kuulemma tulostaa näppäinhatuille merkinnät viime vuodenvaihteessa, mutta vanhasta ruiskuvalukoneesta paljastui öljyvuo- to, joka jätti joidenkin näppäinhattujen pinnan öljyiseksi. Tämä puolestaan pilasi näppäinmerkintöjen tulosteet. Uusin lupaus oli, että ongelman korjaamisen jälkeen näppäinhattuversiot viedään tulostettavaksi toukokuussa ja toimitukset alkavat touko-kesäkuussa 2019. Eivät ole alkaneet.

”Phase 5”-n näppäimistöseikkailuun liittyy vielä toinenkin episodi – vai pitäisikö sanoa toinenkin (anti-)sankari, nimittäin Commodore- ja Amiga-ken- tän legendaarinen lisälaittevalmistelija **Jim Drew** (cbmstuff.com, ex-Utilities Unlimited, EMPLANT). Herra Drew heitti viime vuonna ristiriitaisia tun- teita herättävän nimensä keitokseen ilmoittamalla aikeistaan tuottaa uu- sia Commodore 64 -näppäinhattuja Kiinassa. Hän myös julkaisi hintalaskelmia eri valmistusmaille Lemon64- internet-foorumilla.

Jim Drew tarjosi Michaelille mah- dollisuutta ostaa näppäinhattut hä- neltä ja täten päästä joukkorahoitus- kampanjastaan kunnialla eroon. Jos uskomme Jimin viestiä elokuulta, ja

miksi emme uskoisi, Michaelin vastaus tarjoukseen oli jotakuinkin jäätävä: Jos Jim aloittaa omien näppäinhattujensa ennakkomyynnin, Michael uhkasi vas- tavetona... aloittaa näppäinhattujensa toimitukset. Absurdilla uhkauksella, jos sellainen tapahtui, näyttäisi kuiten- kin olleen toivottu vaikutus: kukaan ei uskaltanut liikahtaa C64:n suuressa näppäinhattusodassa kesän aikana.

Skrollin mennessä painoon syys- kuun alussa Michaelin näppäin- hattuista ei ollut kuulunut mitään. Jim puolestaan lupasi ilmoittaa oman päätöksensä asiasta kuun vaihteessa järjestettyyn CRX 2019 -tapahtumaan (*Commodore Retro eXpo*) mennessä.¹ Sillä välin – mikäli kärsivällisyys tai Retrobright eivät riitä – ratkaisun voi tarjota DS Retro Garagen **DarthSkal- mar**. Hän myy samaisella nimimerkil- lä Ebayssa vanhoja, mutta uudelleen värjättyjä Commodore 64 -näppäin- hattuja/näppäimistöjä.

Yksi asia CRX 2019:ssä oli muuten peräti valmiina: **Evie Salomonin** Back- Bit-moduuli (backbit.io), uusi vaihtoehto C64:n SD-korttiasemaksi (vrt. SD2IEC/EasyFlash, Skrolli 2018.2).

Naputellaan tehokkaasti

Jos pelkkien näppäinhattujen tuotta- minen on vaikeaa, entäpä kokonaan uuden näppäimistön? Tähän kysy- mykseen on vuosien saatossa yrittä- nyt vastata muutamakin Commodore 64 -pienhanke. Yksi on Deskthority.

¹ CRX 2019 tuli ja meni, mutta painoon mennessä korviimme ei ollut kantautunut mitään Jim Drewn näppäinhattuprojektista.

net-käyttäjän **tl** MeC64-projekti vuo- delta 2012, jonka ohjeet ja lähdekoo- dit löytyvät edelleen verkosta (deskthority.net/viewtopic.php?t=4522).

Mainitsin Uusnelonen-artikkelissa myös Breadbox64.comin **MtnBuffa- lon** näppäimistöprojektin, joka tuot- tikin mekaanisen MechBoard64-näp- päimistöprototyypin puolisen vuotta artikkelini jälkeen.

MechBoard64 (breadbox64.com) ei sisällä itse näppäinhattuja, mutta mu- kana tulee kaikki tarvittava uudempien tai vanhempien C64-näppäinhattujen asettamiseksi paikoilleen. Näppäimis- tön ennakkotilauuserä tuli myyntiin ja myytiin loppuun kesällä 2018 – tällä hetkellä odotellaan, jos tai koska seu- raava erä tulee tilattavaksi. Kuten van- han MeC64-projektin kanssa, pientuo- tettujen niche-tuotteiden saatavuus on aina vähän ongelmallista. Kuulopuhei- den mukaan MechBoard64 vaikuttaa kuitenkin ihan kunnolliselta ja kunnii- alliselta projektilta.

Moni muukin on haistanut markki- naraon Commodore 64:n uusionäp- päimistöjen tuotannossa. **Edu Ara- nan** (arananet.net) #C64-näppäimistö ja Kipper2K:n (kipper2k.com) sekä Amiga On The Lake -kaupan (amigaonthelake.com) Kipper2K C64MX ovat tuloillaan – näillä näppäimillä. Näppäimistöt mahtuvat Commodore 64:n kuoriin, mutta mikään näistä projekteista ei varinaisesti yritä toi- sintaa Commodore 64:n näppäimis- tötuntumaa. Ne käyttävät moderneja näppäinmekanismeja.

Mustana hevosenä Commodore 64 -näppäimistöviidakossa laukkaa THEC64

telmäkin on jättänyt draaman päivät taakseen. Hienosti numeroitu MorphOS 3.11 julkaistiin viime vuoden hei- näkuussa, ja käyttöjärjestelmä sai jopa Objective-C-tuen. Sen sijaan kurkistus ”virallisen” Amigan puolelle – jossa vuosikymmenten hajaannus on pilk- konut Amiga-tuotteet ja -oikeudet use- an pienen toimijan tontille – pimentää horisontin välittömästi. Salamoita vis- kelevät Hyperionin ja Cloanton lisäksi kumpaakin lähellä olevat kauppiaat, tekijät ja seuraajat.

Amiga, joka tarkoittaa espanjaksi naispuolista ystävää, ei tarvitse viha- miehiä, koska sillä on vähän liikaa-

kin ystäviä. Kuin hätähuutona viesti Amiga.com-verkkosivulla kysyy edel- leen: ”Do you remember when Amiga meant friend? We do.”

Taistelu Amigasta

On helppo ymmärtää mikä moti- voi AROS- ja MorphOS-kehittäjiä. MorphOS-tiimin suomalaisjäsen **Har- ry ”Piru” Sintonen** vahvisti taannoin arvaukseni tuopin ääressä: on yksin- kertaisesti kivaa tehdä omaa käyttö- järjestelmää. Vaikka Amiga-yhteenso- pivan vaihtoehtoiskäyttöjärjestelmän kehittäminen vuonna 2019 tuntuu silkalta hulluudelta, tekemisen iloa ei

tarvitse selitellä. Skrollille ainakaan. Sen sijaan virallisten Amiga-tuottei- den puolella tapahtuvaa Titanicin kan- situolien uudelleenjärjestelyleikkiä on vaikeampi ymmärtää.

Perusongelma on toki selkeä: His- toriallisten tapahtumien seurauksena Hyperion omistaa lisenssin AmigaOS 4 -käyttöjärjestelmään. Cloanto omis- taa oikeudet AmigaOS 3 -käyttöjär- jestelmään ja vanhempiin versioihin, joihin AmigaOS 4 perustuu. Useat pienemmät tahot omistavat oikeuk- sia keskeisiin komponentteihin, joita AmigaOS 3 ja 4 tai niiden ohjelmistot käyttävät – esimerkiksi Ei näin! -pals-

Och samma på svenska

Olemme Skrollissa käsitelleet paljonkin retrokirjallisuutta – mainittakoon esimerkiksi *Commodore 64: a visual Compendium* ja *Commodore Amiga: a visual Compendium* (Skrolli 2015.3) sekä *Sinclair ZX Spectrum: a visual compendium* (2016.1). Pidempi kirjaluettelomme löytyy osoitteesta skrolli.fi/sisallysruettelo/kirjat.

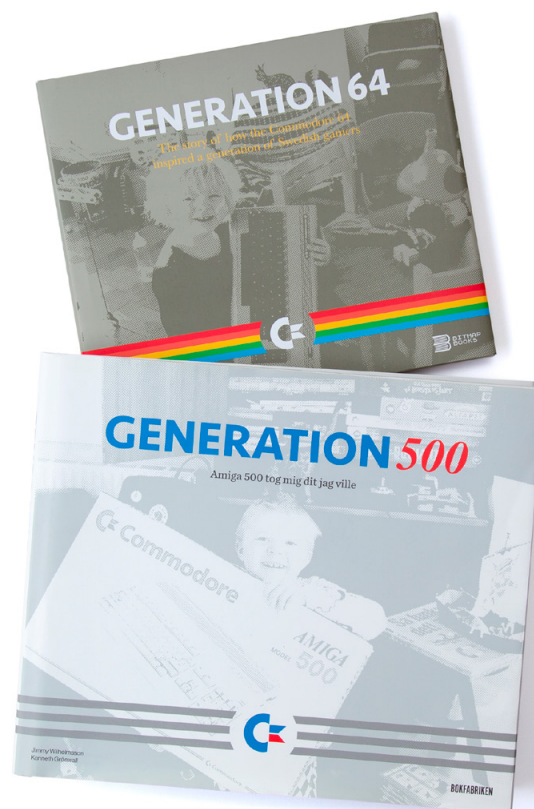
Emme kuitenkaan ole vielä maininneet lehdessä länsinaapurimme Commodore-muistelmia: *Generation 64* (2014) ja *Generation 500* (2017). **Jimmy ”Spelpappan” Wilhelmssonin** kirjoittamat ja **Kenneth Grönwallin** taittamat kirjat kertovat Commodore 64:n ja Amigan tarinan Ruotsissa, joka on luonnollisesti hyvin samankaltainen, kuin niiden tarina Suomessa.

Ruotsalainen kaverini on töissä Commodore-käyttäjillekin tutussa firmassa Elektron – butiikki valmisti Commodore 64:n äänipiiriin SIDiin perustuvaa SidStationia vuosina 1999–2006 (mainitsemisen arvoinen kulttuuriteko Ruotsista tämäkin, jonka noteerasimme jo Skrollissa 2013.1). Kysyin häneltä kirjan tekijöiden taustoista, ja hän kyseli vähän ympäriinsä. Johtopäätös oli lopulta, että tekijät ovat ihan tavallisia Commodore/Amiga-käyttäjiä. Kuvakulmakin on siis tuttu.

Generation-kirjoissa on paljon samaistumisen aihetta meille suomalaisille Commodore- ja Amiga-käyttäjille. Lähin vertailukohta lienee **Juho Kuorikosken** *Commodore 64 - Tasavallan tietokone* -kirja (2017), joskin Generationit ovat visuaalisempia ja kahvipöytämaisempia. Nyt jos koskaan on hyvä syy laittaa pakkoruotsi hyötykäyttöön. *Generation 64* on tosin sittemmin julkaistu päivitetystä muodosta myös englanniksi ja ruotsinkielinen versio on loppuunmyyty, joten tekosyy ruotsinopelle on valmiina.

500- ja 64-kirjojen rakenne on samankaltainen: Runkona on monipuolinen joukko ruotsalaisia henkilöhaastatteluja ohjelmoijista muusikoihin, graafikoista hakkereihin, yrittäjistä lehdistöön. Haastattelujen kautta esitellään eri alakulttuureja ja tekstiä täydennetään aikalaisvalokuvilla. Henkilöesittelyjen välissä on aukeamatolkulla visuaaleja – lehtileikkeitä, mainoksia sekä peli- ja demokuvia. Myös keskeiset historian käänteet ja skenetapahtumat saavat sivutilaa.

Olen monesti mietiskellyt, että retromarkkinoiden parhaita puolia on uusi mahdollisuus kokea niitä asioita, jotka silloin joskus jäivät kokematta. Nyt olisi tilaisuus saada välähdyks Commadore- ja Amiga-kokemuksista länsinaapurissamme – niin samanlaisista, mutta kuitenkin erilaisistakin. *Generation 500* on saatavilla ruotsiksi esimerkiksi Bokus.comista ja *Generation 64*:n englanninkielistä versiota saa muun muassa Amazonista. Kirjoja on nähty myös meikäläisissä kirjakaupoissa.



talla (2019.2) mainittiin nykyisin Individual Computersin omistama P96-grafiikkakirjasto. Lisäksi markkinoilla toimii myös laitevalmistajakumppaneita, kuten **Trevor Dickinsonin** A-EON Technology, joka valmistaa muun muassa AmigaOne-tietokoneita AmigaOS 4 -käyttöön.

Kukin osapuoli on tahollaan käyttänyt aikaa ja rahaa tuotostensa sekä oikeuksiensa kehittämiseen ja hankkimiseen – ja heillä on näille sijoituksille tuotto-odotuksia. Vaikka toisin voisi kuvitella, Amiga-markkinoilla liikkuu kuitenkin vielä hieman löysääkin rahaa. ”Seuratkaa rahaa”, julisti jo klas-

sinen Watergate-elokuva *Presidentin miehet* (1976). Amiga-sijoitukset eivät tosin myöskään ole pelkästään taloudellisia. Vuosikymmenten korpivaelus on rikkonut monia ihmissuhteita, kasvattanut kaunoja ja vaalinut vanhaa vihaa.

Jotkut vuosien saatossa syntyneet oikeudet ovat myös aidosti päällekkäisiä, tulkinnanvaraisia tai riidanalaisia, kuten Hyperionin lisenssi käyttää AmigaOS 3:sta. Hieman edellisen Commodore-sisäpiirini jälkeen, syyskuussa 2018, Hyperion julkaisi AmigaOS 3.1.4 -version (ks. amigastore.eu), jota työstitivät tutut nimet **Olaf Barthel** ja **Tho-**

mas Richter. Vaatimattomasta versionumerosta huolimatta kyseessä on AmigaOS 3.5/3.9:n veroinen päivitys klassikkokäyttöjärjestelmään. Cloanton näkökulmasta kyseinen julkaisu on kuitenkin laitton – tämä kysymys AmigaOS 3:n julkaisu-oikeuksista kun on keskeinen osa parhaillaan käytävää oikeustaistelua Cloanton ja Hyperionin välillä.

Ei näin! -palstalla mainittiin myös Individual Computersin **Jens Schönfeldin** ja Cloanton riita P96-komponentista: Cloantolla on näkemyksensä mukaan jo olemassa oleva lisenssi P96:een, kun taas Individual Compu-



THEC64.

Kuva: Retro Games Ltd.

THEC64-PELIT

Alleykat, Anarchy, Attack of the Mutant Camels, Avenger, Battle Valley, Bear Bover, Boulder Dash, Bouncer, California Games, Chips Challenge, Confuzion, Cosmic Causeway, Cyberdyne Warrior, Cybernoid II, Deflektor, Destroyer, Everyone's a Wally, Firelord, Galencia, Gateway to Apshei, Gribbly's Day Out, Gridrunner (VIC-20), Heartland, Herobotix, Highway Encounter, Hover Bover, Impossible Mission, Impossible Mission II, IO, Iridis Alpha, Jumpman, Mega Apocalypse, Mission AD, Monty Mole, Monty on the Run, Nebulus, Netherworld, Nodes of Yesod, Paratroid, Pitstop II, Planet of Death, Psychedelia (VIC-20), Ranarama, Robin of the Wood, Silicon Warrior, Skate Crazy, Speedball 2, Spindizzy, Steel, Street Sports Baseball, Street Sports Basketball, Summer Games II + Summer Games events, Super Cycle, Sword of Fargoal, Temple of Apshei Trilogy, The Arc of Yesod, Thing Bounces Back, Thing on a Spring, Trailblazer, Uridium, Who Dares Wins II, Winter Games, World Games ja Zynaps.

(thec64.com). Kun esittelin ”Uusnelosta” Skrollissa 2018.1, kerroin myös THEC64 Mini -retrokonsolista ja lupauksista julkaista siitä myöhemmin täysikokoinen versio, jossa olisi toimiva näppäimistö. THEC64 Mini (Skrolli 2018.2) julkaistiin aikanaan ja sai kohtuullisen vastaanoton, mutta fyysisesti kyseessä oli täysin liikahattamaton muovilaatikko.

Vaikka tieto täysikokoisesta THEC64-versiosta näppäimistöineen oli ollut tarjolla alusta saakka, yllätyin silti, kun muun muassa Commodore-oikeuksia omistavan ja lisensoivan italialaisen Cloanto-firman pomo **Michael Battilana** kertoi syyskuussa 2018 Suomessa käydessään, että täysikokoinen versio näppäimistöllä ja parannetulla peliohjaimella todella on työn alla. Jos *tämä* Michael niin sanoo, se on todennäköisesti totta.

Kesäkuussa Retro Games Ltd ilmoitti, että THEC64 ilmestyy joulukuussa 2019. Ulkoisesti kyseessä on THEC64 Minin tapaan vanhemman ”leipälaatikkomallisen” Commodore 64:n kloonin varustettuna nyt toimivalla näppäimistöllä ja mikrokytkin-

peliohjaimella. Sisällä sykkii edelleen emulaattorirautaa ja myös liitännät ovat modernit, joten vanhoja kuusnelosen peliohjaimia ei sentään pääse hyödyntämään. Laitteen takaa löytyy totutusti HDMI-näyttöliitäntä, eikä ilmeisesti juuri muuta.

Päivitetyn pelivalikoiman lisäksi mukana on tietenkin myös Commodore 64:n basic-ohjelmointitila sekä uutuutena VIC-20-tila peleineen ja omine basic-eineen. Alkuperäinen VIC-20 on varsin identtinen vanhemman Commodore 64-kotelon kanssa, joten tämä on mieluisa bonus. Valmiin pelikirjaston (ks. laatikko) lisäksi molempiin tiloihin voi ladata USB:n kautta omiakin pelejä.

THEC64 on edelleen vailla varsinaista Commodore-tavaramerkkilisenssiä, vaikka basic-ROMit onkin lisensoitu. Commodoren konkurssipesän 1990-luvulla ostanut Escom myi Commodore-logon aikanaan hollantilaiselle Tulipille tämän PC-mallistoa varten. Tulip meni konkurssiin 2009, ja tavaramerkki on Battilanan tietojen mukaan ”ilmeisesti sittemmin siirretty yhtiölle Aruballa”.

Aidan toisella puolella

Nähtäväksi jää, millainen näppäimistö THEC64:ään lopulta ilmestyy. Niin tai näin, osataan sitä aidan toisellakin puolella, nimittäin Spectrum-tontilla. Kirjoitin Spectrum-sisäpiirissä 2018 (Skrolli 2018.4) Commodore 64:n aikalaiskilpailijaan, Sinclair ZX Spectrum-tietokoneeseen, liittyvistä sekalaisista retroprojekteista. Yksi näistä oli Retro Computersin kannettava

ters P96:n uutena omistajana katsoo oikeudekseen veloittaa kirjaston käytöstä lisämaksuja. Ratkaisutkaan eivät aina ole ammattimaisia, kun toimijat ovat pieniä ja rahaa vähän: Cloanton kieltäytyttyä Individual Computers ilmoittautui vetäytyvänsä Amiga-markkinoilta kokonaan, ilmeisesti kustoksi.

Käytännössä viralliset Amiga-markkinat ovat kurjistuneet sarjaksi oikeusjuttuja ja niiden uhkia. Esimerkiksi anonymi Amiga Documents -sivusto (sites.google.com/site/amigadocuments, twitter.com/amigadocuments) seuraa ja dokumentoi näitä tapahtumia säännöllisesti – tosin selvästi Hy-

perionin vastaiselta puolelta leirijakoa.

Kysyin vuosi sitten ensimmäisessä Commodore-sisäpiirissäni Cloanton Michael Battilanalta, voisiko asialle tehdä jotain. Diplomaattinen Michael vastasi kauttarantain. Vuosi on taas vierähtänyt ilman, että mikään on olennaisesti muuttunut. Seuraavassa minun arvaukseni.

Fanipojista kasvoi aikuisia

Olen ollut osana useita teknologisia faniyhteisöjä, joissa pyhien lehmien palvontaa on ilmeistä ja noudattaa tuttua logiikkaa. Tutuimmat esimerkit ovat varmasti Apple ja Tesla. Applea fanitet-

tiin pitkään logiikalla **Steve Jobs** ei voi tehdä virheitä. Hohto on **Tim Cookin** valtakaudella himmentynyt – itse pidän Applesta ja Cookista vähän enemmän sen ansiosta. Teslaa fanitetaan edelleen ajatuksella, ettei **Elon Musk** tee virheitä, joskin tässäkin tapauksessa maalipinta on alkanut rakoilla.

Myös Amigaa fanitettiin aikanaan samalla logiikalla. Fanitus ei tosin henkilöitynyt yhtä paljoa, vaan kohdistui ennen kaikkea Amiga-tavaramerkkiin. ”Amiga”-nimellä yritettiin pitkään markkinoida asioita, joilla oli kovin vähän tekemistä oikean Amigan kanssa – niitäkin fanitettiin. Kun viralliselle

pelikone ZX Spectrum Vega+, jonka kohtalona oli jämähtää muun muassa näppäimistöongelmiin, jotka seurasivat keväällä 2016 tapahtunutta projektitiimin hajaantumista.

Ilokseni tai surukseni voin ilmoittaa, että ZX Spectrum Vega+ -projektista nähtiin viimein haamu helmikuussa 2019. Yhtiöstä aikaisemmin hädetyt retropelialan ja Alten8-peliyhtiön vaikuttaja **Paul Andrews** sekä Spectrum-hakkeri ja -kirjailija **Chris Smith** olivat ilmeisesti palanneet projektiin ennen loppua, mutta työsarkaa taisi olla jäljellä enää sen alarajassa. Kaksi eri lähde on vihjannut, että mukana roikkunut Sinclair-perustaja **Clive Sinclair**, 79, on puolestaan jo sairaaloinen ja poissa-olevainen.

Näyttää siltä, että ZX Spectrum Vega+:n surullinen tarina jää noin kymmeneen (Retro Computersin vahvistamaton väite on 400) maailmalle toimitettuun prototyypitasoiseen tuoteseen ja yli puoleen miljoonaan puntaan joukkorahoittajien haaskattuja rahoja. Papereiden mukaan velkaa jäätii muuallekin. Esimerkkinä joukkorahoitusmarkkinoiden arvaamattomuudesta Retro Computersin karu loppu kiskoi julkisuudessa alas myös aiemmin arvostettua nimeä. **Janko Mrcic-Flogel** Private Planetista, joka tunnetaan hyviä arvioita saaneesta Gemini PDA:sta (Skrolli 2018.2), on saanut kritiikkiä osuudestaan Vega+-sotkuun.

Mrcic-Flogel toimi yhtenä johtajana Retro Computersissa. Teoria, johon olen taipuvaisin uskomaan, kulkee jotakuinkin näin: Mrcic-Flogel, sa-

Amigalle ilmestyi teknisesti yhteensopivia, mutta epävirallisia ja muun nimisiä haastajia, niitä paheksuttiin laajalti. Erityisesti kun MorphOS-julkaistiin, Amiga-yhteisö repesi pitkäksi aikaa kahtia punaiseen (virallinen Amiga) ja siniseen (kilpailija) leiriin.

Tällä fanikulttuurilla tai vanhalla kahtiajaolla ei kuitenkaan nähdäkseen ole enää juurikaan merkitystä. Nykyisin virallinen Amiga-leiri kinastelee pitkälti keskenään. Syyt riidoille vaikuttavat myös olevan paljon "aikuisempia" kuin silloin joskus. Tämä on tavallaan hyvä asia. Uskon, että sovinnolle on enemmän edellytyksiä, kun

ZX Spectrum Next. Kuva: specnext.com.

moin kuten Andrews ja Smith, jäivät pahaan välikäteen muuten huonosti hoidetussa firmassa ja joutuivat ulos kukin vuorollaan sen seurauksena. He auttoivat lopussa lähinnä pistämään pillit pussiin, kuten parhaiten osasivat. Vanha herra Sinclair tuskin oli paljoa mukana missään vaiheessa. Mutta ken tietää – ja ne, jotka tietävät, eivät kerro. Yhtiön muu johto, joka jääköön tässä nimeämättä, vaikuttaa olleen puikoissa suurimpien myrskyjen aikaan.

Kirjoitin taustasta Spectrum-sisäpiirissä ja lisää kiistanalaisia väitteitä Retro Computers -yhtiöstä voi lukea kaapatusta osoitteesta retro-computers.co.uk. Myös The Register -verkkolehti on seurannut tapahtumaketjua ansiokkaasti. Joukkorahoitusmaailmasta tulee joskus mieleen vanha sanonta: tie helvettiin on kivetty hyvillä aikomuksilla.

Spectrum seuraavaksi

Lupaavin uusista Spectrum-projekteista edellisen Spectrum-sisäpiirini aikoihin oli uusi ZX Spectrum Next

tunteet ovat laimentuneet ja taloudelliset intressit pienentyneet. On paljon Amiga-tekijöitä ja -harrastajia, joita vanhat jakolinjat eivät enää kiinnosta patkääkään.

Arvelen, että objektiivisesti katsottuna suurimpana kantona kaskessa on AmigaOS 4:n lisensoinut Hyperion. Syytä voimme vain arvailla. Tilinpäätöstietojen perusteella Hyperion on kerännyt kohta parikymmenvuotisen projektin aikana melkoiset velat (kehitystyö on pitkälti ollut ulkoistettua). Kenties yhtiö kokee tästä syystä tarpeelliseksi puolustaa kynsin hampain asemaansa. Oma lisämausteensa on, että

-tietokone. Kyseessä on visio siitä, mitä ZX Spectrum -kotimikro voisi tänä päivänä näyttää, jos Sinclair ja yhtiön suunnittelija **Rick Dickinson** eivät olisi ajautuneet erilleen 1980-luvulla eli jos koneen valmistus ei olisi päättynyt Amstradille – ja päättynyt. Helmi-kuussa 2018 kirjoitin, että Nextin toimitusten piti alkaa tammikuussa, mutta muoviosien viimeistelyssä ja ruiskuvalamisessa oli vielä tekemistä.

Yli vuosi myöhemmin, elokuussa 2019, tilanne ei ole joukkorahoittajien kannalta juurikaan muuttunut: ZX Spectrum Nextiä odotellaan edelleen. Vega+:n valossa olisi helppo sortua kyynisyyteen, mutta Next-tiimi on sentään pitänyt joukkorahoittajia ajan tasalla säännöllisillä ja mielestäni uskottavilla päivityksillä. Verrattuna "Phase 5":n näppäinhattuihin tai Vega+:n myöhempien aikojen kiemurteuihin ero Nextiin on kuin yöllä ja päivällä.

Lyhykäisyydessään, ainakin jos tilannekatsauksiin on uskomisen, Nextiä on viivästyttänyt näppäimistö – mi-

aikaisemmin Hyperionin yritysmuotona oli ilmeisesti VoF eli belgialainen versio avoimesta yhtiöstä. VoF:n omistajat ovat henkilökohtaisesti vastuussa yhtiön veloista.

Cloanton suhteen arvioni on suotuisampi. Minun kirjoissani Cloanton Battilana on nähnyt paljon vaivaa selvittääkseen Amigan oikeudellista tilannetta. Cloanton Battilana on uuden Amiga.com-kotisivun takana ja hän on perustanut hiljan Amiga-oikeuksien keskittämistä varten uuden yhtiön: C-A Acquisition Corporationin. Yhtiö on perustettu marraskuussa 2018 Nevadassa, Yhdysvaltoihin, jossa Cloan-

Päätöntä menoa

Crash (1984–1992, 2018–2019) on brittiläinen ZX Spectrum -pelaajien lehti, joka on sittemmin ilmestynyt pari kertaa myös pelkkänä vuosikirjana. Lisäksi lehti ”eli” hetken *Sinclair User* -julkaisun osana vuosina 1992–1993. Kuten sisarlehti *Zzap! 64*, *Crash* tunnetaan taiteilija **Oliver Freyn** upeista kansipirroksista. *Crash* teki tunnetuksi myös toimittajan nimeltä **Lloyd Mangram**, jonka nimi on epäilyttävän lähellä edesmennyttä golfaaja **Lloyd Mangrumia**.

Lloyd Mangram ei ollut paikallinen **Walu, Nordic the Incurable** tai edes **Nnirvi** (jonka oikeaa nimeä monet epäilevät varmaan edelleen keksityksi – minäkin). Hän ei siis ollut pakinoitsija taiteilijanimen takana, eikä hän ollut myöskään Mangrumin haamu. Lloyd Mangram oli ihan reilusti keksitty tyyppi, jonka tehtävänä oli toimia *Crash*-lehden äänitorvena postipalstalla ja milloin missäkin. Yhteisen nimimerkin turvin useat toimituksen jäsenet pystyivät rau-



Silloin joskus... (Kuva: Archive.org)

hassa kettuillemaan lukijoille. Lloyd Mangramilla oli useampikin palsta, joista yksi oli numerossa 6 (7/1984) alkanut *Lloyd Mangram's Hall of Slime*. Limagalleria noin paasti suomentaen. Limagalleriaan lukijat – usein kouluikäiset pojat, joskus tytöt – lähettivät pelituloksiaan, jotka julkaistiin sitten lehdessä lukijan nimen ja kotipaikan kera. Galleria oli kehystetty fantasiakauhumaaisilla piirroksilla. Lehden esipuheista päätellen kunniagalleria

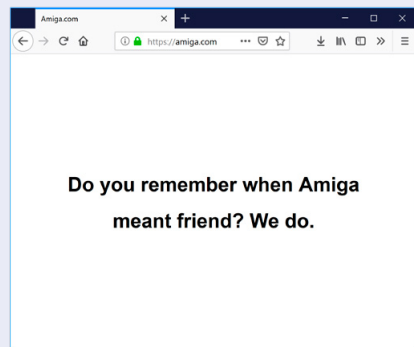


...ja nyt.

oli lehden tekijöiden mielestä vähän vastenmielistä brassailua ja pujausta, joka toteutettiin lukijoiden toistuvista pyynnöistä ja koska sellainen oli tapana – siitä nimi. *Crashin* jouluspesialississa 12 (1/1985) julkaistiin kuitenkin ensimmäinen kaksisivuinen *Valley of Slime*, jossa palstaa elävöitettiin lukijoiden lähettämällä valokuvilla, joista oli leikely eri tavoin teloitettuja irtopäitä. Hiljalleen tapa vaikiintui *Hall of Slime* -palstallakin.

tolla on aiemminkin ollut toimintaa. C-A Acquisition Corporationiin on siirretty Cloanton Amiga Inc.:ltä hankkimat oikeudet, nyt kun vanha Amiga Inc. on viimein lakannut toimimasta. Hyperionia vaivaavan protektionismin sijaan Battilanan tavoite on tunnut jo pidempään olevan museoinnissa ja opetuksessa sekä teknologian avaamisessa paremmin kehittäjien käsiin. Puhetta on ollut jopa Amigan käyttäjärjestelmän ja alkuperäisraudan vapauttamisesta avoimeksi lähdekoodiksi, kun käynnissä olevat oikeusjutut ratkeavat. Toisaalta joidenkin

muiden mielestä Cloanto ja Battilana ovat pelkkiä ”patenttitrolleja”, jotka kiusaavat hyvää tarkoittavia tuotekehittäjiä. Aika näyttää. Individual Computersin P96-protesti lienee seurausta väärinymmärryksistä ja yrityksen omistajan pienrittäjämentaliteetistä, jossa euroja lasketaan hieman turhan tarkkaan ja suurella intohimolla. A-EON Technologyn Trevorilla euroja taas tuntuu olevan ylimääräisiäkin ja hän on tukenut sekä Hyperionin että Cloanton leirejä. Oma vaukseni on, että nämä ja monet muut pienemmät toimijat olisivat houkuteltavissa entistä pa-



Amiga.com rempaan yhteisötoimintaan, jos haajaannus virallisten Amiga-oikeuksien omistajien kesken saataisiin ratkais-

Kuvista muodostettiin vaihtelevan näköinen mestausmaisema yhdistämällä lukijoiden irrotetut päät piirroksien. Oliko se sittenkään ihme, että tietokoneet ja roolipelit tuomittiin kasarilla usein satanismista (Skrolli 2018.4). Mestatut pikkulapset kun tuppaavat näyttämään roisilta.

2010-luvulla ZX Spectrumilla pelanneet pikkulapset ovat fyysisesti kasvaneet aikuisiksi ja palstallekin pääsee enää (joukko)rahalla. Suomalaisena ja commodoristina minulla ei aikanaan ollut tilaisuutta, joten nyt en voinut jättää sitä käyttämättä. Maksoin 35 punttaa (39 €) ylimääräistä Kickstarterissa *Crash*-vuosikirjan tilauksesta päästäkseni palstalle. Ensimmäinen Lloyd, lehtiprojektin nykyinen koordinaattori **Chris Wilkins**, otti pian yhteyttä pyytäen ”sopivaa” kuvaa palstalle. Mikä olisikaan sopivampi kuin Skrolli-kolumnini pelottava asioita pitelevä pää.

Huijarisyndrooma iski viimeistään, kun seuraava Lloyd, yksi alkuperäisen lehden päätoimittajista (ja nykyinen päätoimittaja), **Roger Kean**, otti yhteyttä syksyllä 2018 kysellen ”hi-scoreja” palstalle. Leikittelin hetken *Scuttle Butt* -tuloksillani, mutta lopulta oli todettava, että enhän minä mitään Spectrumilla osaa pelata (sittemmin olen sentään korjannut vajeen *The Hobbitilla*, Skrol-

li 2018.4). Vuodet olivat tehneet tehtävänsä: En ollut ainoa. Kahdelta muultakin päältä puuttuivat pelitulokset kokonaan ja monilla tuloksia oli vain yksi.

Omatuntoni ei silti kolkuttanut kauaa. Tunsin pian jopa jonkinlaista tyydytystä tästä historianlehden käänteestä. Roger Keanista tuli nimittäin alkujaan *Crashin* päätoimittaja alkujaan numerossa 39 (4/1987) – ja samalla hävisivät Limagalleriasta päiden kuvat ja heti seuraavassa numerossa koko palstakin. Alamäki oli alkanut. Jos Roger kehtaa tämän jälkeen, niin kehtaan minäkin...

Limagallerian korvasi aikoihin tylsääkin tylsempi *Robin Candy's* (myöhemmin *Phil King's*) *Scores*. Kahta vuotta myöhemmin pannukakku oli valmis, kun numerossa 63 (4/1989) lukijoiden tuloslistat integroitiin pelien, musiikin ja videoiden myyntilastoihin (!) palstalla *Charts and Scores*. Tätä ”herkkua” kesti kahden numeron verran, kunnes *Crashissa* 65 (6/1989) julkistettu lehtiuidistus karsi palstan ja reippaasti lehden sivujakin rahoittaakseen seuraavan uutuuden: pelikasetit kannessa.

Vanhoja Crash-lehtiä voi lukea Archive.org-verkkokirjastossa: archive.org/details/crash-magazine

tua. Sama koskee monia keskeisiä, itsenäisiä AmigaOS-kehittäjiä, jotka tekevät tarvittaessa töitä kaikkien kanssa.

Loppusota?

Kaikkien näiden vuosien jälkeen virallisen Amigan jäämistö ansaitsi kaukonäköisemmän suojelijan. Oma arvaukseni on, että Amigan tapauksessa Cloanto tarjoaisi tähän todennäköisemmin eväät. Komppaan myös Battilanaa, joka totesi toukokuussa tubettaja **Stephen Jonesin** haastattelussa: ”Pidän MorphOS-tiimistä. Se näyttää esimerkkiä siitä, mitä voidaan

tehdä omilla resursseilla, ilman juridisia ongelmia, omavaraisena tiiminä – ilman kaikkea rumuutta.”

Niin kauan, kun pääpahikset riitelevät, kukin omista syistään, laajemmalta yhteishengeltä puuttuu pohja. Toivotaan, että Cloanton ja Hyperionin välinen oikeusjuttu on se viimeinen taisto, jonka jälkeen Amiga saa rauhan tavalla tai toisella.

P.S. Kannattaa seurata Amiga.comin html-lähdekoodia rot13-salauksenpurkajalla varustautuneena. Viestit vaihtuvat.

käs muukaan. ZX Spectrum Nextin suunnitellut Rick Dickinson piirsi laitteelle modernin perhosnäppäimistön, jollaisia on nähty esimerkiksi Applen kannettavissa tietokoneissa. Ideana kuitenkin oli, että Nextin emolevy toimisi myös alkuperäisen ZX Spectrumin kotelon ja näppäimistön kanssa, joten uuden näppäimistön tulisi olla taaksepäinyhteensopiva.

Alkuperäisen ZX Spectrumin (1982) nähdessään monet ihastelevat arvatenkin ensimmäisenä sen pientä kokoa ja tuskastelevat näppäimistön kumisia painikkeita. Tarkemmin ottaen ZX Spectrumin näppäimistö oli kuitenkin kalvonäppäimistö – kuminäppäinten alla oli kaksi näppäimistökalvoa. Kolmantena asiana ihmetelläänkin sitten pienen koon seurauksia eli näppäimistön omalaatuista asetelua: askelpalautin (*backspace*) on shiftin takana, samoin kaikki nuolinäppäimet. Esimerkiksi nuoli ylös on näppäinyhdistelmä shift+7.

Sinclair korjasi asetteluongelman ZX Spectrum+ -mallissa vuonna 1984, muun muassa lisäämällä näppäimistöön erilliset askelpalautin- ja nuolinäppäimet. Mutta pitääkseen uuden näppäimistön yhteensopivana, yhtiö toteutti nämä uudet näppäimet yhdistelmänäppäiminä (*composite keys*) eli teknisesti alkuperäisen näppäimen ja shiftin yhdistelminä. Näppäimen painallus painoi näppäimistön alla useampaa kalvoa, joista yhdellä kalvolla kytkettiin shiftiä vastaava signaali ja toisella näppäimen signaali (esim. 7, nuoli ylös) – kalvoja oli nyt kahden sijaan kolme kappaletta.

ZX Spectrum -kaapeliyhteensopivuuden säilyttämiseksi myös ZX Spectrum Nextin näppäimistö toteutettiin yhdistelmänäppäimillä. Näppäimistön alla olevien kalvojen määrää tosin laskettiin kolmesta takaisin kahteen, koska kolmikalvoista näppäimistöä pidettiin nykytarpeisiin liian jäykkänä. Näin ollen tekninen ratkaisu pinnan alla oli hieman erilainen kuin ZX Spectrum+ :ssa. Lopulta tämä päätös tulisi viivästyttämään Next-projektia kuukausilla.

Valitettavasti uudet kaksikalvoiset yhdistelmänäppäimet eivät nimittäin toimineet luotettavasti, vaan shift aktivoitui liian usein toisen näppäimen jälkeen, jolloin esimerkiksi nuolesta

ylös tulikin numero 7, mikä on epäsuotavaa. Tilanpäivitystensä mukaan ZX Spectrum Next -tiimi kokeili useita erilaisia kalvopainatuksia ilman tuloksia. Lopulta näppäimistöä varten lisättiin pari uutta signaalilankaa, jotta yhdistelmänäppäimet voitiin toteuttaa perinteisellä tyyllillä. Painoon mennessä jännättiin, onko ratkaisu tässä ja pääsevätkö valmiit tuotteet pian matkaan.

Viivästysten lisäksi muutoksella on pieni historiallinen hintalappu: ZX Spectrum Next oletavasti kelpuuttaa edelleen ZX Spectrumin näppäimistön, mutta sama ei toimi enää toisin päin, koska Nextin näppäimistöä lähtee nyt enemmän piuhoja, kuin vanha Spectrum ymmärtää.

Lisää FPGA-vaihtoehtoja

Edempänä mainittu italialainen Cloanto tunnetaan paitsi Commodore- ja Amiga-oikeuksien nykyomistajana, myös Amiga Forever - (amigaforever.com) ja C64 Forever -emulaattoripaketeistaan (c64forever.com). Molemmista julkaisiin 8. versio joulukuussa. Jo vuonna 2014 verkko-osoitteeseen spectrumforever.com ilmestyi kuitenkin vihjaus seu-

raavasta tuotteesta: ”Work is in progress on a first version of Spectrum Forever, the all-in-one Spectrum emulator suite with games, RP9 authoring and play-back, and more.”

Valitettavasti Battilana kertoi Stephen Jonesin haastattelussa toukokuussa 2019, että ajanpuutteen vuoksi ei-Commodore-pohjaiset Forever-tuotteet ovat jäissä: ”Ajattelimme alkujaan toteuttaa Spectrumin, DOSin, kenties ST:n ja muita... mutta 8-bittinen Commodore ja Amiga pitävät jo kiireisenä.” Hyviä Spectrum-emulaattoreita on toki tarjolla pilvin pimein, mutta Cloanton Forever-emulaattoripaketit ovat huoliteltuja aikakapseleita, joita näkisi mieluummin useammistakin alustoista.

Enemmän tapahtuu onneksi uudelleenohjelmoitavalla FPGA-piirillä varustettujen retrokoneiden saralla. Viime numerossa kerroimme MiSTeristä, universaalista FPGA-retrokonsolista. Myös Spectrum-markkinoille on ilmestynyt FPGA-laitteita. Yksi on taannoinen espanjalainen ZX-Uno (zxuno.speccy.org). Viime maaliskuussa ZX-Unon entisestään kutistettu versio µZX1 ilmestyi Verkami-joukkorahoitussivulle. Rahoitustavoitteesta saatiin kasaan määräajassa vain

alle puolet, mutta tuote on kuin onkin sittemmin ilmestynyt verkkokauppaan (antoniovilena.es).

Olemme kirjoittaneet Skrollissa myös FPGA:lla varustetuista Commodore- ja Amiga-minikoneista, kuten C-One, Minimig ja Turbo Chameleon (Skrolli 2018.2) – viimeksi mainitun toimitukset jatkuivat viimein tammikuussa pitkän tauon jälkeen (icom.de). Kevyesti päivitetty Turbo Chameleon V2 sisältää nyt uutuutena PS/2-varusteliitännät suoraan laitteen kyljessä, kun ennen ne olivat lisäkaapelin päässä.

Myös Commodore 64:n FPGA-vaihtoehtomolevy Ultimate-64 sai pienen päivityksen keväällä: Uusi Ultimate-64 Elite sisältää fyysisiä parannuksia liittimiin ja jumppeista on luovuttu. Emolevyllä on nyt myös toiminto peliohjainporttien vaihtamiseen päittäin. Lisäksi Ultimate-64-kauppaan (ultimate64.com) tuli samoihin aikoihin tarjolle sisäinen kaiutin levyasemaaänien simulointiin. Kuten Gideonin tuotteissa aiemminkin, kannattaa varautua pitkiin tilausjonoihin.

Kuusnelosen tuolla puolen

Kutkuttavin FPGA-parivaljakko ovat kuitenkin edellisessäkin Commodore-sisäpiirissäni mainitsemani ”hemoheisiä vetänyt C128D” C256 Foenix (c256foenix.com) ja ”uus-Commodore 65” MEGA65 (mega65.org). Toisin kuin useimmat aikaisemmat Commodore 64 -retroprojektit – jotka ovat keskittyneet lähinnä korvaamaan vanhan raudan ominaisuudet uudella raudalla – nämä tuovat ZX Spectrum Nextin tapaan mukanaan myös aivan uusia teknisiä kykyjä. Lisäksi MEGA65:n erikoislupaus on tietenkin harvinaisuudeksi jääneen Commodore 65:n herättäminen ”henkiin”.

Molemmista on kuluneen vuoden aikana tihkunut taiseen tahtiin tilannekatsauksia. Kehitystyö Kanadassa ja Saksassa jatkuu selvästi vilkkaana. C256 ei tosin ole pelkkä FPGA-kone, sillä sen ytimenä hyrrää Western Design Centerin 8/16-bittinen



Zzap! 64 ja Crash olivat Newsfield Publicationsin (sittemmin Europressin) julkaisemia sisarlehtiä Commodore 64 - ja ZX Spectrum -pelaajille. Molemmista ilmestyi uudet vuosikirjapainokset tälle vuodelle. Zzap! 64 saa vielä 2020-painoksen syksyllä. Vuosikirjoja voi tilata osoitteesta fusionretrobooks.com.

65C816 (14 MHz) -suoritin. Kyseessä on parannettu versio WDC 65C02:sta, joka taas on kehitelmä MOS Technologyn 6502:sta – Commodore 64:n ja 128:n suorittimien lähisukulaisesta. C256 onkin visio siitä, miltä Commodore 128:n seuraaja olisi saattanut näyttää, jos Commodore ei olisi siirtynyt Motorola 680x0 -pohjaiseen Amigaan.

WDC 65C816 -suorittimen variaatioita käyttävät muun muassa sellaiset legendaariset laitteet kuin Apple IIgs ja Super Nintendo. C256:ssa on kyllä myös uudelleenohjelmitava FPGA-piiri – itseasiassa peräti kolme sellaista. Niiden tehtävänä on tavallaan imitoida Commodoren MOS-erikoispiirejä sekä tietenkin mahdollistaa raudan jatkokehitys ohjelmallisesti.

C256 Foenix -projekti on laajentunut sisältämään kalliimman C256 Foenix FMX -mallin (*Foenix Music Expansion*), jota on paranneltu etenkin musiikintekijöitä silmällä pitäen – siinä on esimerkiksi tupla-SIDien lisäksi paikat SN76489/YM2612-äänipiireille. C256 Foenix FMX:n kehittäjäversio 3D-tulostetulla kotelolla (ks. kuva) on nyt ennakkotilattavissa noin 300–400 dollarin/euron hintaan. Kehittäjäversion toimitusten on luvattu alkavan joulukuussa 2019 tai tammikuussa 2020.

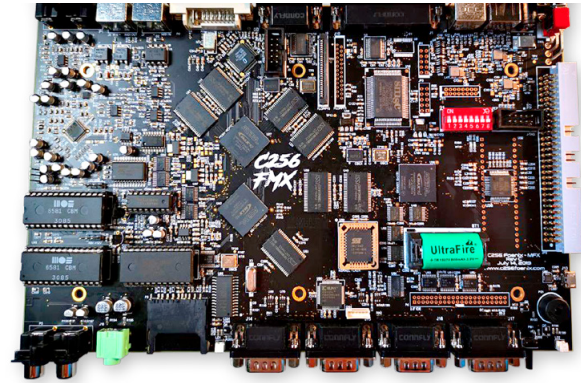
Virallinen Commodore 128:n – tai tarkemmin Commodore 64:n – seuraaja oli tietenkin Commodore 65 (tai C64DX), jota kehitettiin vuosina 1990–1991. Siitä ei kuitenkaan ole tihkunut maailmaan kuin muutamia prototyyppisiä, koska Commodore ei koskaan julkaissut konetta, vaan keskittyi vuoden 1994 konkurssiinsa saakka C64:n jatko tuotantoon ja Ami-

gaan. Museum of Electronic Games & Artin MEGA65 on autenttisen näköinen FPGA-klooni Commodore 65:stä. MEGA65:stä on kehitteillä myös prototyyppiasteella oleva käsikonsoliversio MEGAPhone.

MEGA65:n julkaisu on viivästynyt merkittävästi alkuperäisestä tavoitteesta vuoden 2016 lopulla, mutta kesällä 2019 laitteen uudelleenvalmistettu kotelo ja jopa näppäimistö sekä FPGA-pohjainen emolevy alkavat olla tilannekatsausten mukaan valmiita.

Hyvin Commodore 65:ltä näyttävä MEGA65 oheisessa valokuvassa onkin jo fyysinen laite, ei enää tietokonehallinnus. Kysyin elokuussa MEGAn **Detlef Hastikilta**, koska MEGA65:n julkaisua on lupa odottaa: ”2020 olisi kiva”.

Päysin melkein palstan loppuun puhumatta GEOSista, mutta lopulta on pakko rikkoa lupaukseni. Tärkeänä yhteensopivuuden merkkipaaluna GEOS nimittäin latautui viimein kesällä MEGA65:llä ulkoiselta Commodore 1541 -levykeasemalta. GEOS toimii MEGA65:llä myös natiivisti, jolloin se ennakkotietojen perusteella latautuu alle sekunnissa ja toimii vieläpä varsin korkealla näyttötarkkuudella. Oikein komeaa.



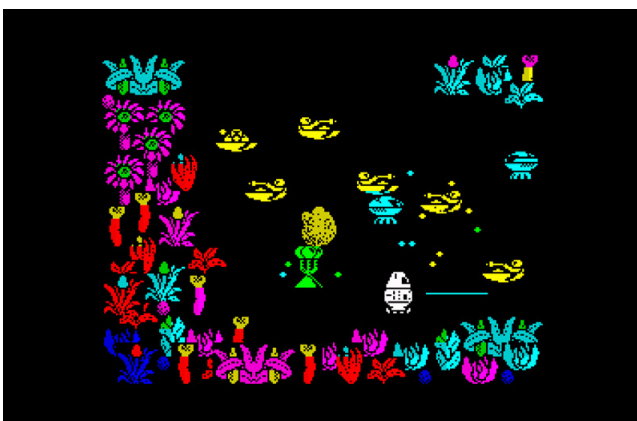
C256 Foenix FMX. Kuvat: Stefany Allaire.

Sekä C256 Foenixin että MEGA65:n kehitystä voi seurata hämmästyttävän yksityiskohtaisesti seuraavista paikoista: C256 Foenix -postituslista (c256foenix.com/subscribe) ja MEGA65:n kehittäjien blogi (c65gs.blogspot.com).

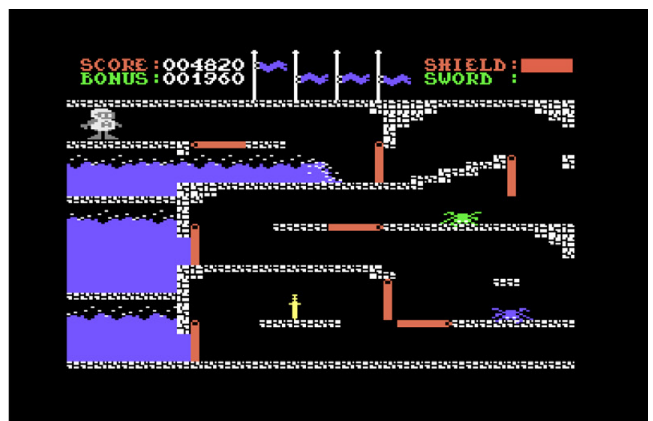
Odottavalla kannalla

Jos C256 Foenix ja MEGA65 kiinnostavat, seurantalistalle kannattaa lisätä myös tuore **David ”The 8-bit Guy” Murrayn, Kevin Williamsin** ja kumppaneiden Commander X16 -emolevyprojekti (facebook.com/groups/CommanderX16, commanderx16.com) – eräänlainen retroilijoiden Raspberry Pi. Laitteen piti perustua samaan suo-

Quadron (ZX Spectrum)



Little Knight Arthur (Commodore 64)





MEGA65. Kuva: © MEGA Museum of Electronic Games & Art, m-e-g-a.org.

rittimeen kuin C256, noin 50–100 dollarin/euron tavoitehintatasolla, mutta tällä hetkellä prototyypin puhutaan käyttävän WDC 65C02:ta ja hintakin voi alkuun lipsua selvästi korkeammalle. Atari-tikkujen sijaan ajatuksissa on käyttää NES-tyylisiä peliohjaimia ja näppäimistö hoidetaan PS/2-liittimellä.

Kun katson kulunutta vuotta taaksepäin Commodore/Spectrum-sisäpiirin näkökulmasta, on todettava, että paljon on tapahtunut – ja samalla juuri mitään (valmista) ei ole tapahtunut. Retromaailmalle tyypilliseen tapaan asiat etenevät harrastelijamaisesti ja verkkaisesti. Lähes kaikkea sitä, mitä odotettiin syksyllä 2018, odotetaan edelleen syksyllä 2019.

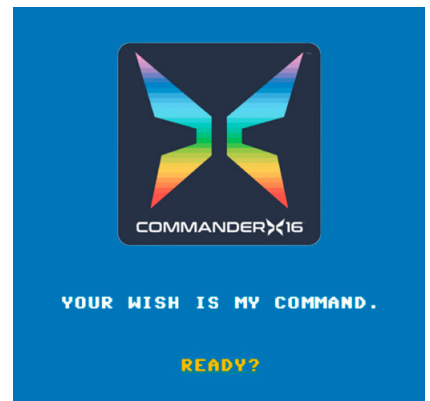
Joskus tuotteen julkaisu todella ottaa aikansa. Vuonna 1988 brittiläinen **Andy ”Cosmium” Beale** viimeisteli ZX Spectrum -pelinsä nimeltään *Quadron*. Pelin julkaisija Palace Software (*Barbarian*) perui kuitenkin julkaisun arvioituaan pelityyppin kysynnän hiipuneen. Viimein 30 vuotta myöhemmin Beale sai Quadronin julkaistua joulukuksi 2018 (cosmium.itch.io/quadron), kasetti: cronosoft.fwscart.com).

Tarina muistuttaa suomalaisen Pasi Hytösen *Little Knight Arthur* -peliä Commodore 64:lle, joka toteutettiin

vuonna 1985 ja julkaistiin vasta 2016 verkossa (pasisbitstuff.net). Joulukuussa 2018 myös *Little Knight Arthur* julkaistiin muuten uudelleen, nyt kasettipainoksena (ka-plus.pl/en/shop/fire-breath-little-knight-arthur).

Ensi kertaan! Palstalle voi kirjoittaa osoitteeseen janne@skrolli.fi. 🏠

Maksuttomat pdf-versiot yli vuoden vanhoista Skrolleista skrolli.fi:ssä: skrolli.fi/numerot



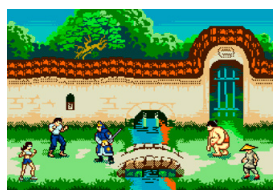
Seuraavalle tasolle!

Vaikka uuden ZX Spectrum Next -tietokoneen (s. 63) varsinaista julkaisua odotetaan vielä, sen Kickstarter-kampanjan ensivaihe toteutui jo vuonna 2017: pelkkien emolevyjen toimitus asiakkaille.

Mikä parasta, laitteesta on julkaistu myös yhteensopiva emulaattori, *CSpect* (cspect.org). Pääset siis tutus-

tumaan koneen kehittyvään *Next-ZXOS*-firmwareen (specnext.com/latestdistro) ja päivitettyihin ominaisuuksiin heti – ja vaikka kehittämään softaa.

Nextissä on esimerkiksi 256-värin tila ja rauta-spriteet, joten uusia pelejäkin on alkanut ilmaantua. Peleihin voi tutustua osoitteessa: www.spectrumnextgames.uk.



EI NÄIN!

SATUNNAISIA SUUTAREITA

Ei näin! -juttusarja sai alkunsa siitä, että minusta on aina ollut mielenkiintoista tutkia, mistä syistä tietyt tuotteet eivät onnistu kaupallisesti, vaikka niissä ei sinällään olisi mitään vikaa. Ja miksi niitä toisaalta tuodaan jatkuvasti markkinoille, vaikka vikaa olisikin. Jaan vahvasti sen näkökulman, että historiallisia virheitä tulee ymmärtää voidakseen välttää niitä jatkossa.

Teksti: Mikko Heinonen

Kuvat: Mikko Heinonen, Mitol Meerna, No Starch Press

Etenkin viimeisen vuoden-puolentoista aikana olen laajentanut aihekirjoa ja kertoillut tarinoita sekä henkilökohtaisen että yleisemmän tietotekniikkahistorian varrelta. Skrollin lavalla Assemblyssä (ks. sivu 86) pitämäni esitelmä kuitenkin palautti mieleeni pari tapausta, jotka ovat jääneet kertomatta tällä palsalla. Palataan siis tällä kertaa juurille.

Kalliiksi käynyt joululahja

Sain viime jouluna ystävältäni, Skrollin taittajanakin tunnetulta **Manulta**, lahjaksi **Evan Amosin** valokuvateoksen *The Game Console*. Amos on jo vuosikausia valokuvannut erilaisia pelilaitteita ja, mikä tärkeintä, julkaissut kuviaan yleiseen käyttöön Wikimedia Commonsin kautta. Tämä arvokas työ on huomattavasti helpottanut esimer-

kiksi näiden omien juttujeni kuvittamista, kun monesta harvinaisestakin laitteesta on ollut valmiina täysin painolaatuisia, vapaasti käytettäviä tuotekuvia. Hänen kirjansa onkin pakollista lukemista kaikille, joita erikoinen pelirauta kiinnostaa.

Käytin joulunpyhät teosta selailen ja havaitsin, että suurin osa sen sisällöstä oli ainakin pintapuolisesti tuttua. Jos en suorastaan omistanut laitetta, olin ainakin lukenut sen tarinasta. Tandy Memorex VISin kohdalla pää kuitenkin löi tyhjää. Päädyinkin paitsi googlaamaan laitetta, myös etsiskelemään, mihin hintaan sellaiset maailmalla liikkuvat. Jälkimmäinen päätös tuli vähän kalliiksi, sillä eBaylle oli vain hetkeä aiemmin listattu toimiva konsoli, jonka hinta poikkesi edukseen aiemmin myydyistä yksilöistä. Se piti tietenkin poistaa. No, joululahjaksi itselle sun muut selitykset.

Videotietojärjestelmä

Lyhytikäiseksi jääneen VISin eli Video Information Systemin tarina liittyy 1990-luvun alun CD-ROM-multimediahuumaan. Romppuhan oli tuolloin talennustilaltaan valtava mutta valmistuskustannuksiltaan suhteellisen edullinen formaatti, jonka odotettiin tuovan myös digitaalista viihdettä yhä lähemmäs peruskuluttajia. Tästä seurasi, että useassa yrityksessä kehiteltiin erityisesti CD-ROMien toistoon suunniteltuja laitteita. Etulinjassa markkinoille saapuivat

muun muassa Commodore CDTV ja Philips CD-i, joita molempia on käsitelty tässäkin juttusarjassa. Tietokonevalmistaja Tandy Corporation halusi myös osansa tästä kakusta, jossa vielä tuohon aikaan vaikutti olevan paljon jaettavaa.

Tandylla oli tuolloin tallissaan monia eri brändejä sekä luonnollisesti kauppaketju Radio Shack, joka oli ollut keskeinen osa sen toimintaa jo 1960-luvulta alkaen. Uusi multimediakeksintö saikin nimekseen Memorex VIS, ja sitä ilmoitettiin kaupattavan ainoastaan Radio Shackeissa. Tämä ei ollut sinällään poikkeuksellista, nimittäin Radio Shack oli noihin aikoihin kenties maailman suurin elektroniikkaketju, ja sillä oli laaja joukko muitakin yksinoikeustuotteita.

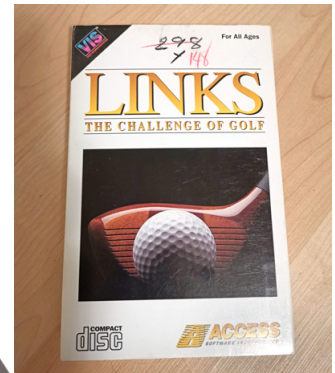
Teknisesti VIS ei ollut erityisen kunnianhimoinen, vaan ammensi sen sijaan luotettavasta tekniikasta ja Tandyn osaamisesta PC-tietokoneiden valmistajana. Tandyn omien laitteiden lisäksi sen tehtaat nimittäin tekivät sopimustuotantoa lukemattomille PC-brändeille, ja Tandy Corporation oli maailman suurimpia henkilökohtaisten tietokoneiden valmistajia.

Vuonna 1992 markkinoille tuodun VISin sydämenä sykki Intel 80286, joka tosin puhui lisälaitteilleen paikallisväylän kautta ja käsitteli muistia ilman odotustiloja. Tämän ansiosta käytännön suorituskyky oli enemmänkin 386SX-mikron tasolla. Muistia koneessa oli megatavun verran, tietenkin tutussa 640+384 kilotavun järjestelyssä, ja sen käyttöjärjestelmänä oli Modular Windows -nimellä tunnettu erikoisversio Windows 3.1:stä. Se sijaitsi yhdessä minimaalisen DOSin kanssa megatavun ROM-piirillä. Cirrus Logicin näytönohjain ja Mitsumin tuplanopeuksinen CD-asema olisivat voineet olla kuin mistä tahansa PC-kloonista, mutta kurioositeettina mainittakoon, että Yamahan piirejä käyttänyt äänikortti on yhteensopiva Adlib Goldin eikä Soundblasterin kanssa. Laitteeseen saa liitettyä PS/2-näppäimistön ja hiiren, mutta pääosin sitä komennetaan infrapunalla toimivilla ohjaimilla. Trendin mukaisesti kapistus näyttää hifi-laitteelta tietokoneen sijaan.

VISko vaikka menemään

Samanlaisen funktion lisäksi Commodore CDTV:llä, Philips CD-i:llä ja Tandy VISillä on itse asiassa yllättävän paljon muutakin yhteistä. Ne olivat





Memorex VISin mukana toimitettiin kaksi infrapunaohjainta. Siihen nähden on erikoista, että varsinaisia kaksinpelejä laitteelle ei oikeastaan edes julkaistu.

kaikki markkinoille tullessaan tehoitaan melko vaatimattomia, hinnaltaan kalliita ja kaupallisesti epäonnistuneita. VIS on kuitenkin tässäkin seurassa vielä omilla lukemillaan, sillä konsoleita arvioidaan myydyin vain kymmenisentuhatta kappaletta ennen kuin Tandy lopetti leikin kesken. Myyjiltä liikanimen ”Virtually Impossible to Sell” saaneet laitteet saivat lähtöpässit Radio Shackien hyllyiltä vuoden 1994 alussa. Modular Windowsia ei enää koskaan käytetty missään laitteessa, vaikka sitä oli suunniteltu yleisemminkin televisioon kytkettävien laitteiden käyttäjärjestelmäksi. Tandy myi koko Memorex-nimen eteenpäin jo vuonna 1993.

Jälkiviisaasti on helppo nähdä, miksi VISejä ei suorastaan revitty käsistä. Muiden valmistajien tapaan Tandy oli yliarvioinut yleisen multimedian houkuttelevuutta ja aliarvioinut vetävän pelitarjonnan merkitystä. Pelaajat kun ovat usein sitä joukkoa, jotka ovat valmiita maksamaan upeista elämyksistä hieman korkeammankin hinnan. VISin pelitarjonnan kärkipäätä edustivat Apple Macintoshilta tuttu klikkailuseikkailu *The Manhole* ja golf-peli *Links*, joita kumpaakin saattoi pelata myös muilla alustoilla. Yli kymmenvuotiaan suorittimen vauhdittama VIS ei kyennyt kovinkaan vauhdikkaaseen toimintaan, eikä infrapunaohjaukseen siihen varsinaisesti rohkaissut. Paketin

päälle lätkäisty lähes 700 dollarin hintalappu oli melkoinen, sillä vielä vuotta myöhemmin samaan hintaan julkaistua 3DO:ta pidettiin tähtitieteellisen kalliina, vaikka se edusti VISiin verrattuna uusinta tekniikkaa.

Suuren osan VISin ohjelmakirjastosta muodostavat erilaiset lapsille ja yleisemmin opetuskäyttöön tarkoitettut multimediaromput. Niiden kääntäminen Windows 3.1:n kokoversiosta VISille lienee käynyt helposti, mutta voidaan kysyä, miksi kukaan olisi valinnut toiminnoiltaan hyvin rajallisen VISin, kun samaa sisältöä saattoi käyttää myös PC-yhteensopivalla tietokoneella.

VISin hylkiöstatus on periytnyt keräilijäpiireihinkin, eikä kone ole juuri lainkaan harrastettu. Multimediaromppuja pyörii kauppapaikoilla muutaman kympin hintaan, jolla ne eivät juurikaan vaihda omistajaa. Yhteensä 21 ohjelman kokoelmaa on yritetty kaupata kuukausikaupalla, hintaa hissukseen alas hinaten. Itse maksoin koneesta ohjaimien ja yhden opetusrompun kanssa noin 150 euroa ja päälle *Manholesta* ja *Linksistä* yhteensä lähes saman verran, mitä tosin pidin jo hieman vastuuttomana omallakin mitapuullani. Kirjoitushetkellä eBayllä on myynnissä on vain yksi kone, josta pyydetään romppukasan kanssa 10 000 dollaria tekstillä ”\$10,000 or \$10 idk offers”.

Saavuttamaton unelma

Siinä missä Memorex VIS on yhä mahdollista poistaa itselleen kohtuuhintaan, samaa ei voi sanoa RDI Halcyonista. Joissakin lähteissä maailman harvinaisimmaksi konsoliksi tituleerattua laitetta valmistettiin vain muutama kappale, ennen kuin taustayritys RDI Video Systems kaatui konkurssiin. Laitteella on tästä huolimatta oma mielenkiintoinen historiansa, joka ansaitsee tulla kerrotuksi.

Rick Dyer tunnetaan ennen kaikkea maineikkaan *Dragon's Lair* -kolikkopelin konseptin suunnittelijana. Harvempi kuitenkin tietää, että *Dragon's Lair* oli tarkoitettu vain välietapiksi, joka mahdollistaisi paljon kunnianhimoisemman seikkailupelin suunnittelun. *Shadoan*-nimellä tunnettu peli ammensi *Sormusten herran* maisemista, ja lohhariluolan yksinkertaisen suuntaohjauksen sijaan tarjolla piti olla huomattavasti enemmän mahdollisuuksia vaikuttaa tapahtumiin.

Dyer suunnitteli jättipeliään alkujaan kelattavien nauhojen ympärille, mutta kuvalevyt osoittautuivat paremmin tarkoitukseen sopiviksi. RDI Halcyon -konsoli koostuikin levysoittimesta sekä sitä ohjaavasta, Z80-pohjaisesta mikrotietokoneesta. Pelin mukana toimitettiin ROM-moduuli, jonka sisältämän ohjelman perusteella levyltä soitettiin oikea animaatiopätkä.

Halcyonin erikoisuuksiin lukeutui myös puheohjaus. Näin digitaalisten assistenttien kulta-aikana se ei ole mitään erikoista, mutta vuonna 1985 tilanne oli hieman toinen. Laite osasi tunnistaa sanat ”yes” ja ”no”, muut sanat kunkin pelaajan piti opettaa sille. Eikä tekstiä tarvinnut itse lukea ruudulta, sillä Halcyonin erillinen text to speech -yksikkö osasi ääntää sen englanniksi. Puheen lisäksi pelien ohjaukseen käytettiin kalvonäppäimistöä.

Se oli kaksi tonnin seteliä

Halcyon siis oli kaikenlaista, mutta melko pian kävi selväksi, että se ei ollut halpa. Tällaisen unelmatekniikan paketointi kuluttajalle myytäväksi vuonna 1985 oli haastavuudeltaan kertaluokkainen projekti, ja pian kävikin selväksi, että kustannukset ovat karkaamassa käsistä.

Halcyonin piti alunperin käyttää RCA:n valmistamia CED-kuvalevyjä. CED oli hieman vinyylilevyn kaltainen, neulaa käyttävä tallennusratkaisu, joka kulutti levyä ja rajoitti sen toistokerrat noin viiteensataan, mutta sen hyviä puolia olivat edullinen valmistuskustannus ja kohtuullisen suuri kapasiteetti. RCA kuitenkin veti CED:n markkinoilta kesken Halcyonin kehityksen, jolloin laite piti suunnitella uusiksi käyttämään

Laserdisciä. Se oli periaatteessa kulumaton ja kuvanlaadultaan parempi formaatti, mutta myös kalliimpi ja tallennustilaltaan pienempi. Peleihin sopivan, CAV-tyyppisen LD-levyn yhdelle puolelle mahtuu vain puoli tuntia videota CED:n tunnin sijaan, joten yhden levyn sijasta tunnin peli vaatisikin nyt kaksi. Myös itse LD-mekanismi oli hintavampi kuin vastaava CED.

Toinen ongelma liittyi mullistavaan ohjaustapaan. Ääni-ohjaus ei ollut reaaliaikainen vaan vaati, että pelaaja puhuu mikrofonin tietyllä hetkellä, minkä jälkeen peli prosessoi ääninäytteen. Tämän myötä *Dragon's Lairin* tyyppiset toimintapelit olivat poissa laskuista. Sen sijaan Halcyonille kehitettiin *Shadoan*-konseptin pohjalta peli nimeltä *Thayer's Quest*, joka oli hieman kuin *DL*, mutta hidastempoisempi ja seikkailullisempi. Sen ohjaukseen käytettiin etupäässä näppäimistöä.

Kaiken tämän lopputuloksena RDI Halcyonin hintalappuun olisi kirjoitettu 2 195 dollaria, nykyrahassa noin 5 000 euroa. Lähes kaikki sidosryhmät ymmärsivät, ettei järjestelmästä tähän hintaan tehdä kauppooja kuin korkeintaan öljysheikkien kanssa. RDI Video Systems yritti kehittää lisärahoitusta myymällä *Thayer's Questia* pelihal-

leihin päivityksenä *Dragon's Lair* -kabinetteihin, mutta lopulta sijoittajien huumorintaju loppui ja julkaisu kuivui kasaan ennen toteutumistaan.

Onnekkaita sattumia

RDI Halcyonilla ei oikeasti ollut menestymisen mahdollisuuksia. Se oli kuin jonkinlainen messuilla hehkutettava proof of concept -tason teknologia-ademo, joka kuitenkin haluttiin tuoda markkinoille, vaikka tekniikka ei ollut valmista. Laitteen hienoutta ei kiistänyt juuri kukaan, hintalapun sitäkin useampi.

Onneksi kaikki tehty työ ei kuitenkaan mennyt hukkaan. *Dragon's Lair*, niin synonyymi epärealistiselle reaktioteskille kuin se onkin, on ansainnut paikkansa videopelien historiassa. *Thayer's Questia* sai lopulta myös ison kaupallisen julkaisun PC-yhteensopivilla, 3DO:lla ja CD-i:llä nimellä *Kingdom: Far Reaches*. Itse en ole peliä toistaiseksi pelannut, mutta sitä kehuutaan ainakin FMV-pelien mittapuulla monipuolisiksi. YouTubesta löytyy useampikin läpipeluuvideo. Jatko-osa *Kingdom II: Shadoan* ilmestyi vuonna 1996, mutta tähän toistakymmentä vuotta aiemmin syntyneen megakonseptin tie päättyi.

Rick Dyerillä oli kovan luokan visio, mutta ehkä vähemmän kaupallista osaamista. Markkinavoimien ehdoilla kokoon puserrettu Halcyon, jonka nimikin tuli *2001: Avaruusseikkailun HAL 9000* -tietokoneesta, olisi varmasti ollut vähemmän scifiä, mutta kokonaisuutena myös tylsempi keksintö. 🐉



Vanhat vampyyrit

Kreivi Dracula on hirviö, joka vain pahenee vanhetessaan. Mutta miten sujuvat ikääntyvien pikselidraculoiden seikkailut?

Teksti: Jukka O. Kauppinen

Kuvat: Jukka O. Kauppinen, Mobygames



```

What shall I do now?
get p
Don't know what "p" is.
What shall I do now?
get up
I'm in a bedroom.
Open window. Brass bed. Sheet tied to
bed.
North
OK
It's getting DARK outside!
What shall I do now?
eat tablet
OK
I'm real PEPPY now!
It's getting DARK outside!
What shall I do now?
CARRYING: Unlit torch. Sixfur matches. 2
small holes in my neck. 2 nozoz tablets.
Pack of Transylvanian cigarettes. Dusty
cloak of garlie.
It's getting DARK outside!
What shall I do now?
  
```

The Count

Scott Adams / Adventure International, 1979
Apple II, Atari 8-bit, Commodore 64, VIC-20, jne.

Seikkailupelien pioneeri **Scott Adams** loi uransa alkuvaiheilla monta teosta, jotka valoiivat pohjaa interaktiivisen fiktion ja koko seikkailupeligenren tulevaisuudelle. Kaikista teoksistaan hän ei kuitenkaan ollut järin ylpeä jälkikäteen. *The Count* nimittäin käsitteli sen verran saatanallisia aiheita, että myöhemmin vahvasti uskoon tullutta Adamsia sen muisteleminen nolottaa, jopa hävettää.

Se on kovin harmillista, sillä vuonna 1979 kirjoitetuksi peliksi *The*

Count on kaikessa yksinkertaisuudessaan ihmeen monipuolinen. Vaikka tekstiseikkailu käyttää vain kahden sanan mittaista parseria eli komentotulkkia, tekstiseinien takana piilee yllättävän elävä maailma. Jokainen pelikerta on erilainen, sillä peli arpoo tapahtumien järjestyksen ja pelin sisäiset tapahtumat etenevät aina eri tahtiin. Pelaajan pitäisi ehtiä tutkimaan Draculan linnaa kilpajuoksussa kelloa eli auringonlaskua vastaan, löytää salakäytäviä ja keinoja, joilla vampyyrilordi pidetään kirjaimellisesti poissa niskasta siihen saakka, kunnes käsissä on työkalut vereniimijän nujertamiseksi. Seikkailussa ei myöskään ole yhtä ainuttä tiettyä ratkaisutapaa, vaan tarina voidaan saattaa kunnialliseen päätökseen monin eri keinoin.

The Count on tiettävästi ja todennäköisesti kaikkien aikojen ensimmäinen videopeli, joka käsittelee kreivi Draculaa. Olkoon se siis myös osa pelisivustystämme.



Dracula

Zen Room / CRL Group, 1986
Commodore 64, Amstrad CPC, ZX Spectrum

CRL koki karvaan pettymyksen julkaistessaan *Dracula*-kauhuseikkailun vuonna 1986. Brittityhtiö oli näet ehkä ensimmäinen pelitalo, joka hoksasi kohumarkkinoinnin merkityksen videopelien myyntikäyriin. Niinpä firma yritti kovasti lobata pelilleen Ison-Britannian elokuvatarkastamon K18-ikärajaa, siis aikana, jolloin videopeleillä ei edes ollut ikäsuosituksia tai omaa ikärajajärjestelmäänsä. Pieleen meni, peli sai vain K15-lätkän. Seuraavaan

peliiin laitettiin vielä verisempiä kuvia. Silloin viimein onnasi ja peli pääsi kohulehtiin!

Dracula oli kuitenkin oiva yritys ja hyvä pelikin, vaikka se jäi historiantekijöihin parhaimmin markkinointikeinoistaan. Se oli myös varsin laaja teos, sillä tekstiseikkailu jaettiin kokonaiseen kolmeen osaan. Koko tarina kun ei mahtunut kerrallaan kuusnelosen ja muiden aikansa hittimikrojen muistiin. **Bram Stokerin** teokseen pohjautuvassa tarinassa seikkaillaan kolmessa esikuvateoksen tärkeimmässä tapahtumapaikassa ja yritetään selvittää Draculaisen juonia, jotka johtavat pelaajan lopulta öisen Lontoon verisille kaduille.

Yllättävää kyllä, *Dracula* painottaa enemmänkin tunnelmointia ja tarinankerrontaa. Toisin kuin monissa aikansa tekstipeleissä, *Draculassa* ei ole panostettu mahdottomassa määrin kieroihin puzzleihin ja ongelmanratkontaan. Tarina kulkee vähän kuin itsestään, mikäli nyt tekstipelistä niin voi sanoa. Eihän tämä mikään seikkailupelien suurin klassikko ole, mutta tärkeä pala pelialan ja genrensä historiaa kuitenkin.



Brides of Dracula

Toast Dept. / Gonzo Games, 1992
Amiga, Atari ST

Draculan elämä on joskus kovin yksinäistä. Ajatella, siellä se hiipparoi kivilinnassaan ja vampyroi naapurikylien asukkaita – mutta kenen kanssa kreivimme istahtaa yhteiselle aamupalalle juttelemaan työstään ennen päivänille siirtymistään? Niinpä. Kaukaisen ja kylmän Transilvanian linnan seinien sisällä ulvoo yksinäisyys. Niinpä kreivi päättää hommata vaimon, ja kun toimeen ryhdytään, niin ryhdytään kunnonla. Hommataan kerralla 13 vaimoa!

13 vaimoa Draculalle, 13 vampyyrintappoasetta Van Helsingille, siinä *Brides of Draculan* ydin. Kun pelifirmakin on nimeltään Gonzo Games, niin eihän tämä voi olla kuin silkkaa

kovuutta. *Spy vs Spyn* tapaan jaetulla ruudulla pelattava yhden tai kahden pelurin vampyyriseikkailu on niin tuhtia B-luokkaa, että se suorastaan hurmaa. En voi kuin ihaila animaattorin näkemystä Van Helsingin ja Draculan kävelytyylistä – tai pelisuunnittelijaa, jonka mielestä Dracula hurmaa naisia vaimokseen puremalla seikkailun lomassa löytyviä povipommiblondieja. Saati sitä, että nämä muuttuvat kertapuraisusta Elviroiksi ja astelevat nöyrästi linnan kellariin arkkuun käyttöään odottamaan.

Yllättävintä on kuitenkin se, että vaikka peli onkin kökkösarjaa, niin se on toimivaa ja viihdyttävää kökköä. Pelimaailmassa on vilinää ja vilskettä. Molemmille sankareille voi käydä kehnosti myös ympäristön toimesta, jos vaikkapa vampyyri erehtyy törtöilemään keskellä päivää kyläläisten keskelle. Tässä on vahvaa hoopoilu-peli-iltaman potentiaalia, kunhan ei vain odota kovinkaan nopeatempoista tai perinteisen viihdyttävää elämystä. Ja massiiviset propsit tekijöille näin omintakeisesta Dracula-tulkinnasta.



The Twisted Tales of Spike McFang

Red Company / Bullet-Proof, 1993
SNES

Jos Draculan vaimot herättivät konseptillaan hämmennystä, samoin teki myös japanilainen vampyyritoimintaseikkailu *McFang*. Se on JRPG eli japanilainen toimintaroolipeli, jossa pelaaja on muun muassa zombielauvoja ja ilkeitä ihmisiä vastaan taisteleva nuori vampyyriprinssi. Aika siistiä – peli, jossa vampyyrit ovatkin hyviksiä ja oikeasti pelattavia päähenkilöitä!

Voihan herra McFangin vampiirisuudesta olla vähän epäileväinen, sillä näin kyvyttömiä vampyyreitä ei ole ihan joka päivä nähty. Itse ainakin muuttuisin mieluummin lepakoksi

ja lentäisin vierivien tukkien ja muiden esteiden yli kuin hyppisin itseni näännyksiin. Nuorukaisen muissakin aktiviteeteissa on hieman epävampyyrimäisiä piirteitä: hän muun muassa nauttii taisteluissa kaatamiensa vihollisten sydämet tuoreina ja verisinä. Käteviä ensiapupakkauksia!

Vaan niinhän japanilainen käsitys vampyyreistä onkin aivan omanlaisensa. Sikäläinen kulttuurihan ei edes tuntenut koko vampyyrien käsitettä ennen kuin se saapui toisen maailmansodan jälkeen amerikkalaisten elokuvien myötä. Ei siis ihme, että sikäläiset vampyyrielokuvat ja -pelit ovat välistä aika vänsiä. Joten tosi iso käsi tälle pinkeälle japanioudolle, joka toivottavasti ilahduttaa SNESin JRPG-pelien ystäviä.

** BONUS LEVEL **

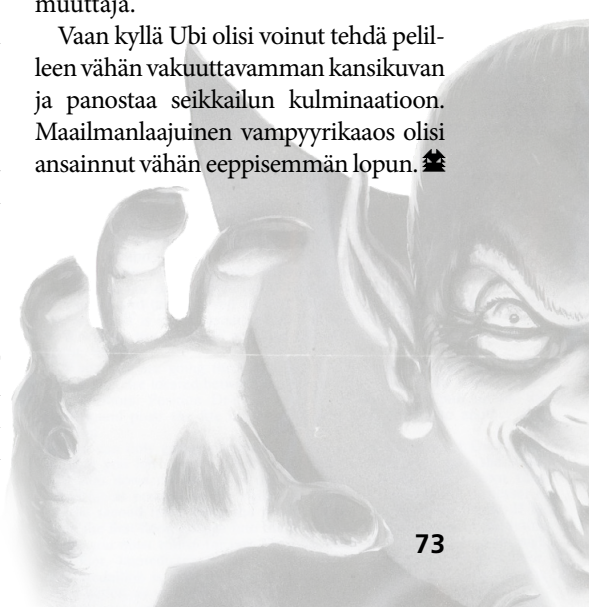


Night Hunter

Ubi Soft, 1988
Amiga, Atari ST, Amstrad CPC, ZX Spectrum, DOS

Vampyyrit kuvataan peleissä liki aina pahiksina ja vastustajina, mutta Ubi Softin kulttiteoksessa *Dracula* on kerankin ehdoton päähenkilö, kunnon sankari. Tavoitekin on ylevä: herra haluaa luoda Maan päälle lopullisen maailmanlopun kaaoksen. Kaikki eivät tästä tykkää, joten *Dracula* on paitsi kova taistelija, myös kyvykäs muodonmuuttaja.

Vaan kyllä Ubi olisi voinut tehdä pelilleen vähän vakuuttavamman kansikuvan ja panostaa seikkailun kulminaatioon. Maailmanlaajuinen vampyyrikaos olisi ansainnut vähän eepisemmän lopun. 🦇





Scorched Earth silloin ja nyt

Muistoja palaneilta mannuilta

Teksti: Raine Liukko

Kuvat: Raine Liukko, Janne Sirén

1990-luvun tykkipeliklassikko peluuttaa edelleen – nyt vuorossa yksi tarina siitä, miten legenda pidetään elossa sekä pelillisesti että maantieteellisesti.

Verkkomonipelit olivat lähipiirissäni 1990-luvulla ja 2000-luvun alussa harvojen huvia. Useamman yhtäaikaisen pelaajan osallistuminen edellytti usein saman koneen ääreen kokoontumista. Pienessä huoneessa, jonka jaoin veljeni kanssa, sisäilma heikkeni peli-istunnon edetessä, mutta tunnelma oli katossa kun taivaalle singottiin Death's Head.

Peli oli *Scorched Earth*, joka meidän peliporukassa tunnettiin tuossa vaiheessa nimellä tykkipeli. Pari vuotta sitten vanha peliporukkamme palasi jälleen tykkipelin äärelle.

Poltettu maa

Scorched Earth on alunperin PC:n MS-DOS-käyttöjärjestelmälle vuosien

1991–1995 aikana kehitetty vuoropohjainen peli, jossa kaksiuolotteiseen maisemaan sijoitetut ihmis- tai tietokoneohjatut tankit yrittävät eliminoida toisiaan yksin tai joukkueittain. Pelaajia voi olla 2–10 kappaletta. Tähtääminen tapahtuu piipun kulmaa ja laukaisutehoa säätämällä.

Vastustajien tuhoamisesta ja vahingoittamisesta tienaa rahaa, jolla voi ostaa erien välillä aseita ja muita tarvikkeita. Tekijän kotisivut löytyvät edelleen osoitteesta www.whicken.com/scorch, mistä pelin shareware-versio on ladattavissa, sekä kaikki viralliset versiot sisältävä kokoelmapaketti *Totally Scorched* omavalintaiseen hintaan.

Perusidea on yksinkertainen, mutta pelin suola on sen tarjoamien asetusten, aseiden ja varusteiden monipuolisuus. Asetuksilla voidaan vaikuttaa muun muassa aseiden, fysiikan, kenttien ja tankkien toimintaan. Lista ominaisuuksista on pitkä, mutta esimerkiksi ammusten aiheuttamien räjähdysten koko, maaston epätasaisuus ja painovoima ovat säädettävissä. Jopa tuulelle voidaan määrittää maksimi-voimakkuus, ja se voidaan asettaa muuttumaan laukaisuvuoroittain.

Peliasetuksista on myös valittavissa, ampuvatko tykit ammuksensa samaan aikaan, kun kaikki pelaajat ovat ammuksensa tehneet, vai vuorotellen pelaajan vuoron päätteeksi.

Runsaudensarvi

Monipuolinen tarjonta jatkuu aseissa ja varusteissa, joita tankkeihin voi menestyksekkäistä sotatoimista ansaituilla varoilla hankkia. Pelin alussa käytettävät varat ovat myös määritettävissä. Perusaseena tankeista löytyy *Baby Missile*-nimistä ammusta voi suoltaa taivaalle niin kauan kuin vuoroja riittää.

Monista ammuksista, tai paremmin sanottuna ammustyypeistä, on usein 2–3 eri versiota. Niiden nimeämiskäytäntö noudattelee pääosin mallia, jossa ammustyypistä on vaatimattomien *Baby*-etuliitteellä oleva versio, ilman etuliitettä oleva välimalli sekä suurin, kaunein ja usein myös kallein *Heavy*-etuliitteen omaava tehokkain paukku. Kevyintä mallia myydään yleensä 10 kappaleen erissä, välisarjan mallia 3–5 kerrallaan ja raskaimmat yksittäin.

Oletusasetuksilla *Heavy*-versiot ovat aina luokkansa kalleimmat edustajat. Pelistä löytyy kuitenkin myös niin kut-

suttu Free Market -tila, jossa aseiden ja tarvikkeiden hinnat vaihtelevat kysynnän perusteella. Markkinoissa voidaan ottaa myös huomioon aikaisemmat pelikerrat, jolloin omissa peleissämme on syntynyt tilanteita, joissa esimerkiksi kolmen kappaleen erissä myytävä Baby Nuke on noussut hinnaltaan yksittäin myytävän isoveljensä Nuken ohi. (Markkinahistorian nollaaminen onnistuu tarvittaessa poistamalla tiedosto SCORCH.MKT.)

Perinteisten räjähdysvoimaan nojaavien paukkujen ohella valikoimasta löytyy myös maaston hävittämiseen tai luomiseen tarkoitettuja ammuksia. Maaston hävittämiseen soveltuvat ammuksia, esimerkiksi Digger, on tarkoitus kohdistaa tankkien sijaan maastoon, yleensä toki suhteellisen lähelle kohteena olevaa vihollista. Ne toimivat hyvin esimerkiksi mäkien päälle sijoittuneisiin vastustajiin – olettaen tietenkin, että pelin asetuksissa painovoima on määritetty käyttöön. Onnistunut osuma maastoon hävittää maata vihulaisen alapuolelta, joka romahtaa alas. Tämän seurauksena tankki parhaassa tapauksessa tuhoutuu tai ainakin vaurioituu vaarattomaksi. Vaurioituminen laskee nimittäin myös käytävissä olevaa laukaisuvoimaa.

Tarvikevalikoimasta löytyy tankkien ympärille muodostettavia, eri vahvuisia ja vastustajien ammuksien osuimiin eri tavoin reagoivia suoja. Tämän lisäksi löytyy laskuvarjoja ehkäisemään vaurioita aikaisemmin kuvatussa tilanteessa, kun maa-aines yllättäen häviääkin telaketjujen alta. Menovettä tankeissa ei oletukse-

na ole, mutta jos sotakoneet tahtoo saada liikenteeseen muutenkin kuin painovoimavusteisesti, myös polttoainetta on myytävänä.

Rehellisyyden nimissä on kuitenkin todettava, etteivät pelin tankit edusta maasto-ominaisuuksiltaan edes 1990-luvun sotateollisuuden huippua, joten etenkin mäkisimmässä ympäristössä liikkuminen on usein enemmän uhka kuin mahdollisuus. Vaurioituneesta tankista operointiin on tarjolla lisävirtaa akkujen (Battery) muodossa, jotka ovat myös edellytys joidenkin tarvikkeiden ja aseiden hyödyntämiseen.

Tykkiä peliin

Pelaajat voivat valita tankin (tai tykin) seitsemästä eri vaihtoehdosta. Erot ovat pääosin visuaalisia, mutta osasta malleista uupuu telaketjut, eikä niillä näin ollen pysty myöskään liikkumaan. Yksi tankeista ampuu Baby Missileita ja Missileita kolme ammusta kerrallaan. Viimeksi mainittu malli on shareware-versiossa saatavilla vain rajoitetusti tietokoneen ohjastamille pelaajille. Totally Scorched -paketin mukana tulevassa ja vanhoissa pelin

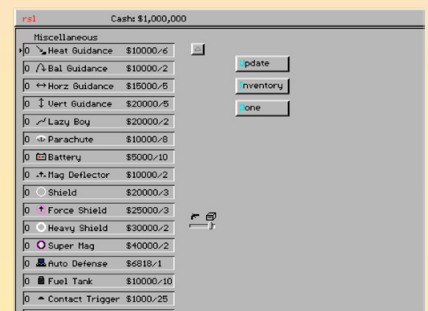


Uutta peliä aloittaessa määritellään pelaajien nimet ja tankit.

rekisteröidyissä versioissa tykki on vapaasti käytettävissä.

Omassa lapsuudessani pelin monipuoliset asetukset jäivät suurilta osin hyödyntämättä kielitaidon puuttuessa. Välillä kuitenkin sattumanvaraisella yritys-erehdys-menetelmällä onnistuimme saamaan enemmän ja vähemmän toivottuja muutoksia pelikentälle. Onneksi asetukset eivät oletuksena talletuneet, joten pelin uudelleenkäynnistys pelasti tilanteissa, joissa ei enää ollut niin selvää, minkä valikon takaa arvoja oli muutettu.

Peli tukee näppäimistön lisäksi myös hiirtä, mutta pelaaminen onnistuu yhtä lailla pelkällä näppäimistölläkin – ainakin teoriassa. Tykkipelikoneessamme ei ollut hiirtä, ja osittain tästä syystä pelasimme peliä viikkoja ostaen vain kaupassa ensimmäisellä sivulla näkyviä aseita. Pelikokemus saikin aivan uusia ulottuvuuksia, kun joku keksi että page up ja page down -painikkeilla voi kaupassa olevia tuotteita selata ja aseita löytyi lisää. Se oli kuitenkin vasta alkusoittoa sille, kun löysimme tarvikevalikoiman tabulattorin takaa. Tämän jälkeen meni tovi jos



Vasemmalla pelin asevalikoimaa. Ensimmäisessä tykkipelikoneessamme näytön resoluutio ei riittänyt kaikkien aseiden listamiseen samalle sivulle. Yllä tarvikevalikoimaa.

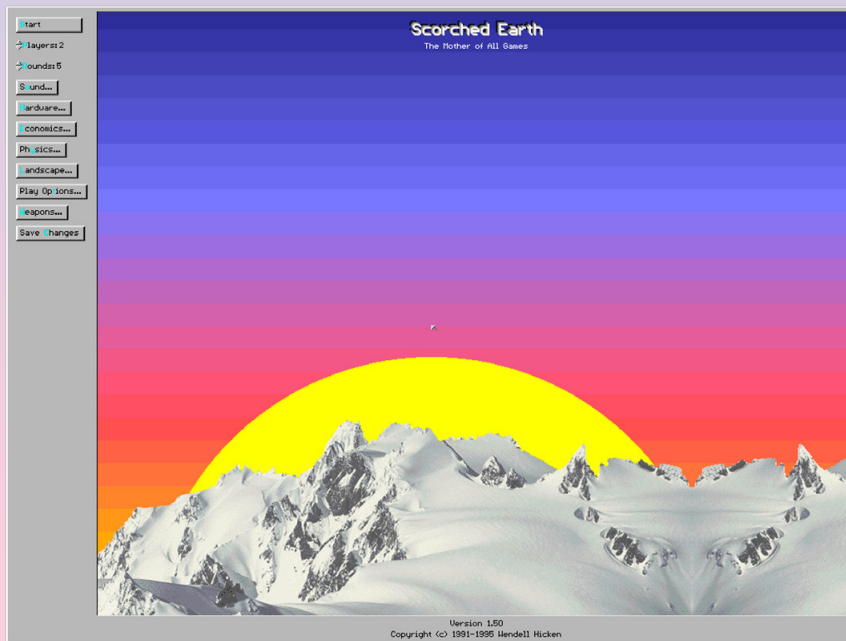
toinenkin selvittäessä, miten ostettuja tarvikkeita voi käyttää tai mitä apua niistä ylipäättään on.

Muksuna pelatessamme taktiset elementit eivät ehkä olleet se merkillepantavin osa tekemistä. Siihen asti että löysimme Economics-valikon takaa asetuksen, jolla voi määrittää paljonko rahaa on alussa käytettävissä, suurin tavoite oli saada kerättyä valuuttaa sen verran, että kauppaan ilmestyi Death's Head -ase. Huuto äityi välillä kovaksi, kun ruudun ylälaudassa kentän yläpuolella lentävien ammuksien sijainti-indikaattori alkoi hidastaa tahtia oman tankin lähetyvillä, mitä seurasi hetken päästä naurunremakka, kun yhdeksän Baby Nuken kokoista räjähdystä täytti ruudun. Puhumattakaan niistä PC-piipparin tuottamista mykistävästä räjähdysäänistä. Toisinaan hihkuttiin siitä vahingonilosta, kun viimeisillä pennosilla ostettu paukku viuhui ylimitoitetun laukaisutehon seurauksena jonnekin pelikentän ulkopuolelle.

Scorchio!

Mukana voi olla ripaus nostalgiahuuma, mutta ainakin meidän peliporukkaamme Scorched Earth on jaksanut viihdyttää vielä nykyäänkin. Grafiikkoja on turha lähteä vertailemaan nykypeleihin, mutta omasta mielestäni ne ovat riittävän hyvät, eikä pelattavuus oleellisesti kärsi niiden vuoksi. Niinpä olemme palanneet Scorched Earthin pariin vielä aikuisiälläkin.

Ei peli tosin ollut mitään silmäkarkkia enää 1990-luvun viimeisellä puoliskollakaan, saati 2000-luvun alussa, kun sitä meillä vielä nuorempana tahkottiin. Äänien osalta nostalgikiintiö täytti parin ensimmäisen Funky Bombin jälkeen. Pelin varsinainen idea kuitenkin kantaa hienosti vielä 2020-luvunkin häämötäessä, kiitos monipuolisten asetusten, aseiden ja tarvikkeiden, jotka mahdol-



Scorched Earth -päävalikko.

listavat hyvin erilaisia lähestymistapoja tankkien välisiin kohtaamisiin.

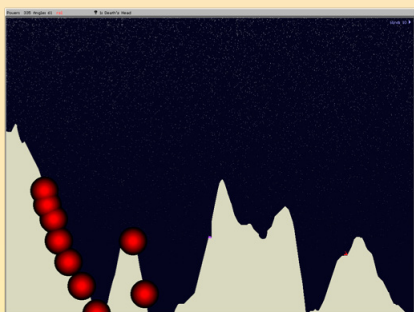
Olemme välillä hakeneet peliin myös uutta näkökulmaa rajoittamalla aseet minimiin, minkä lisäksi on erikseen sovittu, että tietyt aseet (mm. Sandhog) jätetään ostamatta. Olemme määrittäneet Synchronous moden (Play Options -> Mode -> Synchronous) käyttöön, jolloin tankin ampuvat yhtäaikaaisesti, kun kaikki pelaajat ovat määrittäneet laukaisusäätökset.

Pelin pahin puute omasta mielestäni onkin se, että tankit sijoitetaan pelaajamäärästä riippuen vakioituille paikoille vaaka-akselilla tarkasteltuna. Näin ollen pelissä on suhteellisen helppo opetella tietyt piipun kulmat ja laukaisuteho, joilla osuu niin sanotusti riittävän lähelle. Etenkin jos käytössä on kentällä vieriviä pommeja Rollereita, muuttuva tuulikaan ei välttämättä vaikeuta hommaa riittävästi. Rajoittamalla asevalikoimaa olemme saaneet ainakin hieman korostettua osumatark-

kuuden merkitystä. Tämä on kuitenkin vain yksi tapa lähestyä ongelmaa – laaja tarvikevalikoima jättää edelleen tilaa luovuudelle.

Edellä kuvatuilla asetuksilla pelatessamme maa-aineksen lisäämiseen tarkoitettut Dirt-pommit ovat nousseet taktikoinnin kannalta merkittävään asemaan. Toisinaan niillä pelataan aikaa ampumalla niitä suojaksi oman tankin niskaan tai blokataan vastustajan ampumalinja tilanteessa, jossa vihulaisen koordinaatit ovat paremmin selvillä kuin omat.

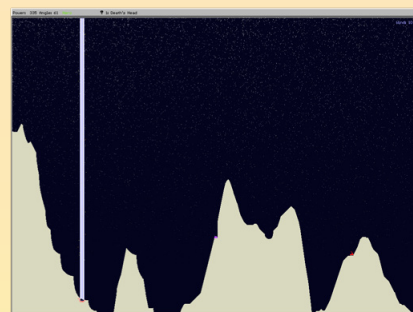
Muiden aikuisuuden mukanaan tuomien pelaamista häiritsevien tekijöiden ohessa vanha ja osittain päivittynyt Scorched Earth -peliporukkamme oli vuosien saatossa levinnyt (myös) maantieteellisesti. Tämä hankaloitti huomattavasti saman näppäimistön ääreen palaamista, joten oli kehitettävä muita ratkaisuja tykkipeli-istuntojen mahdollistamiseksi. Se onkin sitten oma tarinansa, jonka voit lukea viereisestä artikkelista. 🌸



Death's Head -ase tositoimissa.



Funky Bomb saattaa koitua kohtalokkaaksi myös ampujalleen, mutta viihdearvo ainakin on taattu.



Tankki lähtee tuhoutuessaan milloinkinlaisen animaation saattelemana.

Monipeliklassikko verkkoon

Näin jatkat Scorched Earth -matseja ja muita saman koneen ääressä pelattavia vuoropohjaisia pelejä etänä verkossa.

Teksti: Raine Liukko

Kuvat: Raine Liukko, Janne Sirén

Kun saman koneen ääreen palaaminen ei ollut aina mahdollista, ensimmäiset Scorched Earth -etäpelimme tapahtuivat TeamViewer-nimistä etäkäyttösovellusta hyödyntäen. Toisin sanoen, joku ajoi peliä koneellaan ja jakoi työpöytänsä muille pelaajille. Homma toimi, mutta pelitapahtuman aloitusta ei voinut luonnehtia kovin suoraviivaiseksi operaatioksi. Lisäksi käytettävyydessä ilmeni ajoittain pieniä haasteita, ja ratkaisu häiritsi oleellisesti peliä isännöivän henkilön koneen muuta käyttöä.

Näin kypsä ajatus pula-ajan Scorched Earth -verkkomonipelistä, jonka kikkailin käyttöön. Toteutukseni on ruma, eikä kovin tietoturvallinen. Näin ollen on syytä tässä kohtaa huomauttaa, ettei se ole soveltuva sellaisenaan käytettäväksi julkisessa internetissä. Tutun ja luotettavan kaveriporukan kesken sillä sen sijaan pelaa vuoropohjaisia pelejä, jos niille ei parempaakaan verkkopelitoteutusta ole saatavilla.

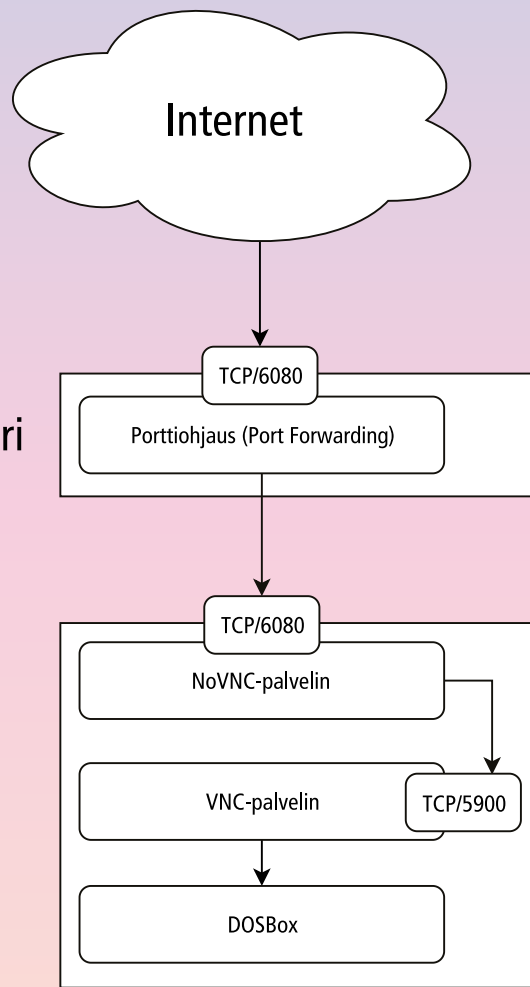
Usean osallistujan reaaliaikaisia toimintoja vaativiin peleihin ratkaisussa syntyy todennäköisesti liikaa viivettä. Myöskään äänet eivät tässä toteutuksessa kuulu kuin mahdollisesti isäntäkoneesta. Ideamme toimii kuitenkin yllättävän hyvin Scorched Earthin kaltaisten vuoropohjaisten pelien kanssa. Myös esimerkiksi Skrollin numerossa 2018.4 esitellyn MHM2000:n pelaamiseen ratkaisu olisi arvatenkin oikein riittävä.

Näennäinen LAN

Alkuperäinen toteutuksemme koostui Linux-jakelua (Fedora) pyörittävästä virtuaalikoneesta, DOSBox-sovelluksesta, tietenkin itse pelistä Scorched Earth, X11vnc-palvelisovelluksesta, noVNC-sovelluksesta ja Nginx-palvelimesta.

Reititin/palomuuuri

Virtuaalikone (esim. Fedora)



Havainnekuva Scorched Earth -verkkopeliratkaisusta.

Paletti toimii myös ilman Nginx-palvelinta, mutta omassa ympäristössäni se sattui olemaan valmiiksi asennettuna. Päätin hyödyntää sitä toteutuksessa välityspalvelimenä, niin kutsuttuna reverse proxyna. Nginx tai välityspalvelimet olisivat kuitenkin jutun aihe jo itsessään, joten tässä artikkelissa kuvaan ainoastaan hieman yksinkertaistetun toteutuksen.

Verkkoliittymän suhteen oleellista on, että reitittimellä on käytössä oma julkinen IP-osoite tai että osallistujat muodostavat keskenään esimerkiksi yksityisen virtuaalisen verkon (Virtual Private Network/VPN). Viimeksi mainittu olisi myös tietoturvanäkökulmasta parempi vaihtoehto. Virtuaalisessa yksityisessä verkossa muodostetaan nimen mukaisesti verkko, mihin tässä tapauksessa peliä isännöivä kone ja osallistujakoneet liittyvät Internetin yli.

Näin saadaan luotua tilanne, jossa koneet ovat näennäisesti samassa lähiverkossa. Tällaisia verkkoja voi suhteellisen helposti muodostaa esimerkiksi Hamachin (www.vpn.net) tai

ZeroTierin (www.zerotier.com) kaltaisilla sovelluksilla. Tämäkin aihe olisi oman juttunsa arvoinen, joten tässä artikkelissa oletan peliä isännöivän koneen olevan yhdistetty Internetiin julkisen IP-osoitteen omaavan reitittimen takaa, mikä on suhteellisen yleinen tapa kiinteissä internet-liittymissä.

Näennäinen Linux

Kuten edeltä voi lukea, toteutuksessa on tehty jo monta kompromissia tietoturvan suhteen, joten suosittelen luomaan ainakin erillisen virtuaalikoneen pelin ajamista varten. Jos virtualisointi ei ole kovin tuttua tai virtuaalikoneita ei ole aikaisemmin luonut, on Oraclen maksuton VirtualBox (virtualbox.org) suhteellisen helppo tapa aloittaa. Ensimmäisen asennetaan virtuaalikone halutulla Linux-jakelulla. Tässä artikkelissa kerrotut ohjeet toimivat ainakin Fedoralla.

Virtuaalikoneelle ei tarvitse varata kovin paljon tehoja. Alkuperäisessä toteutuksessa käyttämäni kone oli käyttöjärjestelmäversionsa puolesta jo auttamatta vanhentunut, joten tätä

artikkeliä ja sen ohjeiden todentamista varten asensin uuden koneen, johon tuli yksi vCPU, 2 Gt keskusmuistia ja yksi 10 gigatavun levy. Verkkosovittimen liitin samaan verkkoon, jonka kotiverkon reititin näkee, jotta artikkelissa myöhemmin ohjeistetut porttiohjaukset (Port forwarding) toimivat.

Fedoran asennuksessa kannattaa valita asennettavaksi vähintään jokin yksinkertainen työpöytäympäristö, koska se on edellytys esimerkiksi DOSBoxin ajamiselle. Software Selection-määritysten alta kohdassa Base Environment laitoin rastin valintaan Basic Desktop ja viereisessä sarakkeessa kohtaan LXDE, joka asentaa em. nimisen melko kevyen työpöytäympäristön.

Kun Fedoran asennus lataa ja asentaa tarvittavia paketteja, voi samalla määrittää salasanan pääkäyttäjälle (root) ja luoda ns. normaalin käyttäjän. Nämä toimenpiteet kannattaakin tehdä. Normaalin käyttäjän nimellä ei periaatteessa ole väliä, mutta käyttäjätunnukseen viitataan tämän artikkelin ohjeissa käytetyissä poluissa. Esimerkikoneessani käytin tunnusta tykki.

Näennäinen DOS

Kun virtuaalikone on valmis, koneelle tarvitaan sovellukset DOSBox, X11vnc ja Novnc. Asennus ja muut toimenpiteet on suoraviivaisinta tehdä komentoriviltä. Artikkelissa kuvatulla asennusvaihtoehdolla mukana tulee oletuksena LXTerminal-niminen terminaaliemulaattori, jonka pitäisi myös löytyä työpöydän vasemman alalaidan kuvakkeista. Asennus tapahtuu root-käyttäjänä, joten siirrytään pääkäyttäjätilaan komennolla:

```
su
```

Tämän jälkeen asennetaan edellä mainitut sovellukset komennolla:

```
dnf install dosbox x11vnc novnc git -y
```

Seuraavat toimenpiteet eivät edellytä enää root-käyttäjän oikeuksia, joten palataan takaisin normaaliksi käyttäjäksi näppäinkomennolla CTRL+D tai komennolla:



NoVNC:n aloitusruutu, jossa kysytään mm. VNC-yhteydelle määriteltyä salasanaa.

```
exit
```

Tämän jälkeen voidaan ladata Scorch Earth tekijän kotisivuilta:

```
wget http://www.whicken.com/scorch/scorch15.zip ~/
```

Luodaan pelille oma hakemisto käyttäjän kotihakemistoon:

```
mkdir ~/scorch
```

Peli on pakattu, joten puretaan se luotuun hakemistoon:

```
unzip ~/scorch15.zip -d ~/scorch
```

DOSBox on MS-DOS-emulaattori, joka mahdollistaa DOS-sovellusten ajamisen myös uudemmissa käyttöjärjestelmissä. Se ei toimi sellaisenaan Linuxissa komentorivillä vaan vaatii graafisen ympäristön, mutta ohjelman käynnistäminen DOS-sovellusten ajamiseen onnistuu näppärästi komennolla dosbox <sovelluksen polku>. Nyt kun sovellus on ladattu ja purettu virtuaalikoneelle, peli lähtee käyntiin seuraavasti:

```
dosbox ~/scorch/SCORCH.EXE
```

Kun peli oli käynnissä, pitää kehittää kanssapelaajille etäyhteys siihen. Ensimmäinen ajatus oli käyttää jotakin VNC-pohjaista (Virtual Network Computing) toteutusta ja asennuttaa kanssapelaajille VNC-asiakasohjelmat. Tähän ideaan tartuinkin, vaikka pelaajien ei lopullisessa ratkaisussa enää tarvinnutkaan asentaa itse asiakasohjelmia. Sovellukseksi valikoitui

X11vnc. Se tuli ensimmäisenä vastaan etsiessäni VNC-sovellusta, jolla onnistuisi helposti jakaa vain yksittäinen ikkuna olemassa olevalta istunnolta. Tämän lisäksi sovellus löytyy Fedoran oletussäilöstä, ja sovelluksen ajaminen onnistuu suoraan komentoriviltä.

Koska halusin koko työpöydän sijaan jakaa vain DOSBox-sovelluksen ikkunan, jossa Tykkipelin aloitusruutu jo näkyi, täytyi selvittää kyseisen ikkunan yksilöivä id. Se onnistuu sovelluksella xwininfo, joka kannattaa ajaa myös komentoriviltä. Id saadaan selville komentamalla sovellus käyntiin:

```
xwininfo
```

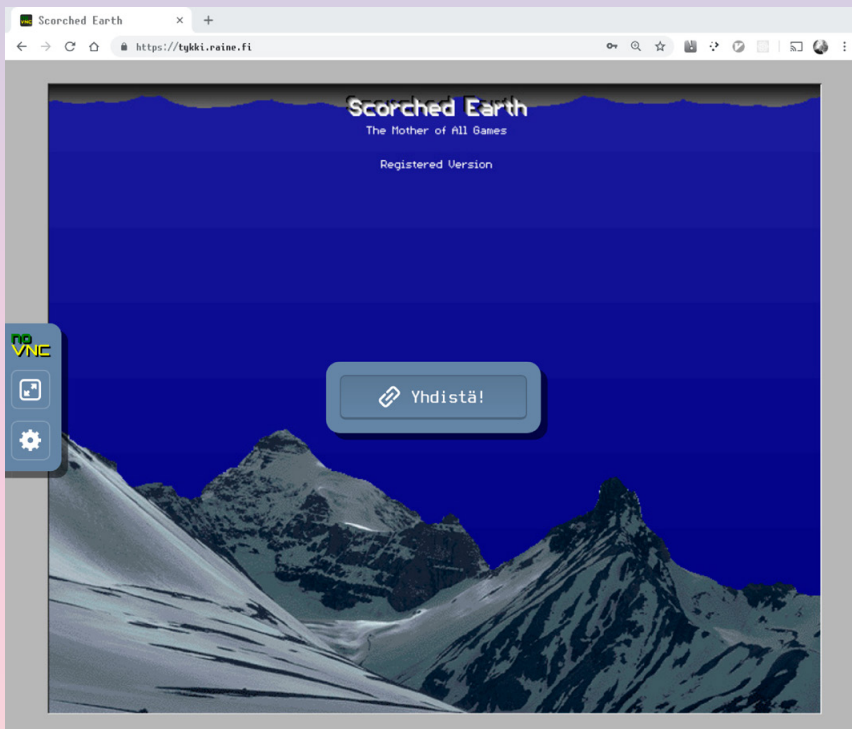
Tämä jälkeen tarvitsee vain klikata DOSBox-sovelluksen ikkunaa ja xwininfo kirjaa tukun tietoja siihen liittyen. Ikkunan id on ilmaistu seuraavaan tapaan:

```
xwininfo: Window id: 0x3400007
```

Halusin kuitenkin luoda skriptin, joka käynnistää pelin ja luo etäyhteydet valmiiksi, joten tarvitsin saman tiedon suoraan komentoriviltä ilman välivaiheen klikkailuja. Xwininfo tarjoaa myös tähän ratkaisun, ja pienellä grepin ja awkin yhteistyöllä pelkän DOSBox-ikkunan id:n saa seuraavasti:

```
xwininfo -root -tree | grep DOSBox | awk '{ print $1 }'
```

Haluamme suojata tulevat VNC-istunnot salasanalla, joten ennen ensimmäistä VNC-palvelimen käynnistä-



Jossain vaiheessa, kun mopo oli jo lähtenyt keulimaan, piti muokata vielä NoVNC:n aloitusruutukin teemaan sopivaksi!

tyskertaa luomme salasana-tiedoston seuraavalla komennolla:

```
x11vnc -storepasswd
```

Komento kysyy salasanan kahteen kertaan ja tallentaa sen oletuksena käyttäjän kotihakemistoon tiedostoon `.vnc/passwd`. Tämän jälkeen VNC-palvelinsovelluksen voi pyöryttää käyntiin alla näkyvällä komennolla, jossa ikkunan id-tietona on käytetty aikaisemmassakin esimerkissä käytettyä arvoa.

```
x11vnc -sid 0x3400007 -shared
-loop -forever -rfbauth ~/.vnc/
passwd
```

Asetus `-shared` mahdollistaa useamman samanaikaisen client-yhteyden. X11vnc myös oletuksena sulkee istunnon, kun asiakasovellus katkaisee yhteyden, mutta asetus `-forever` kumoaa toiminnon. Peliin aikana huomattiin myös ajoittaista epävakausta X11vnc:n toiminnassa, minkä vuoksi lisättiin asetus `-loop`, joka käynnistää sovelluksen uudelleen, jos se jostain syystä terminoituu. Asetuksella `-rfbauth` kerrotaan polku käytettävään salasana-tiedostoon.

Näennäinen client

Vastaan putkahti myös selainpohjainen toteutus VNC-asiakasovelluksesta nimeltään noVNC (github.com/novnc/novnc). Näin ollen ajatus jokaisen

pelaajan erikseen asennettavasta asiakasovelluksesta sai lentää romukoppaan. NoVNC löytyy Fedoran mukana tulevasta asennuslähteistä. Mukaan on paketoitu pieni web-pavelin ja websocket-proxy siltä varalta, että VNC-palvelinsovelluksesta ei löydy suoraan tukea websocketille.

Alkuperäisessä toteutuksessa liikenne tuli salattuna reverse proxy-välityspalvelimelle asti, mutta jatkoi omassa sisäverkossani salaamattomana NoVNC:lle. Koska esimerkkitoetutuksessa ei ole välityspalvelinta ja NoVNC tukee myös salausta, se kannattaa ottaa käyttöön. Tätä varten tarvitsemme avainparin, mikä tässä yhteydessä tarkoittaa yksityistä avainta (private key) ja sertifikaattia.

Koska sertifikaatti on allekirjoitettu itse luodulla avaimella, se ei ole oletuksena luotettu peliin osallistuvien koneiden käyttöjärjestelmän tai selaimen toimesta. Tämän vuoksi peliin yhdistäessä tulee varoitus epäluotettavasta sertifikaatista. Tämän tulisi normaalisti soittaa käyttäjän hälytyskelloja, mutta tällä kertaa emme anna asian vaivata. Seuraavalla komennolla luomme yksityisen avaimen ja sen pohjalta vuoden voimassa olevan sertifikaatin:

```
openssl req -x509 -newkey
rsa:4096 -nodes -keyout ~/avain.
pem -out ~/tykki.pem -days 365
```

Komentoa ajettaessa kysytään useita asioita; tällä kertaa oleellisin asia on Common Name, johon kannattaa määrittää liittymän julkinen IP-osoite tai DNS-nimi. Jos liittymän IP-osoite vaihtuu usein, kannattaa harkita esimerkiksi dynaamisesti päivittyvän DNS-palvelun käyttöä. Se helpottaa peliin osallistumista, kun yhdistäminen tapahtuu aina samalla nimellä, eikä mahdollisesti vaihtuvalla IP-osoitteella.

NoVNC edellyttää, että avain ja sertifikaatti löytyvät samasta tiedostosta, joten yhdistämme ne komennolla:

```
cat ~/tykki.pem ~/avain.pem > ~/
cert.pem
```

Tämän jälkeen NoVNC-palvelin voidaan käynnistää komennolla:

```
novnc_server --vnc 127.0.0.1:5900
-cert ~/cert.pem
```

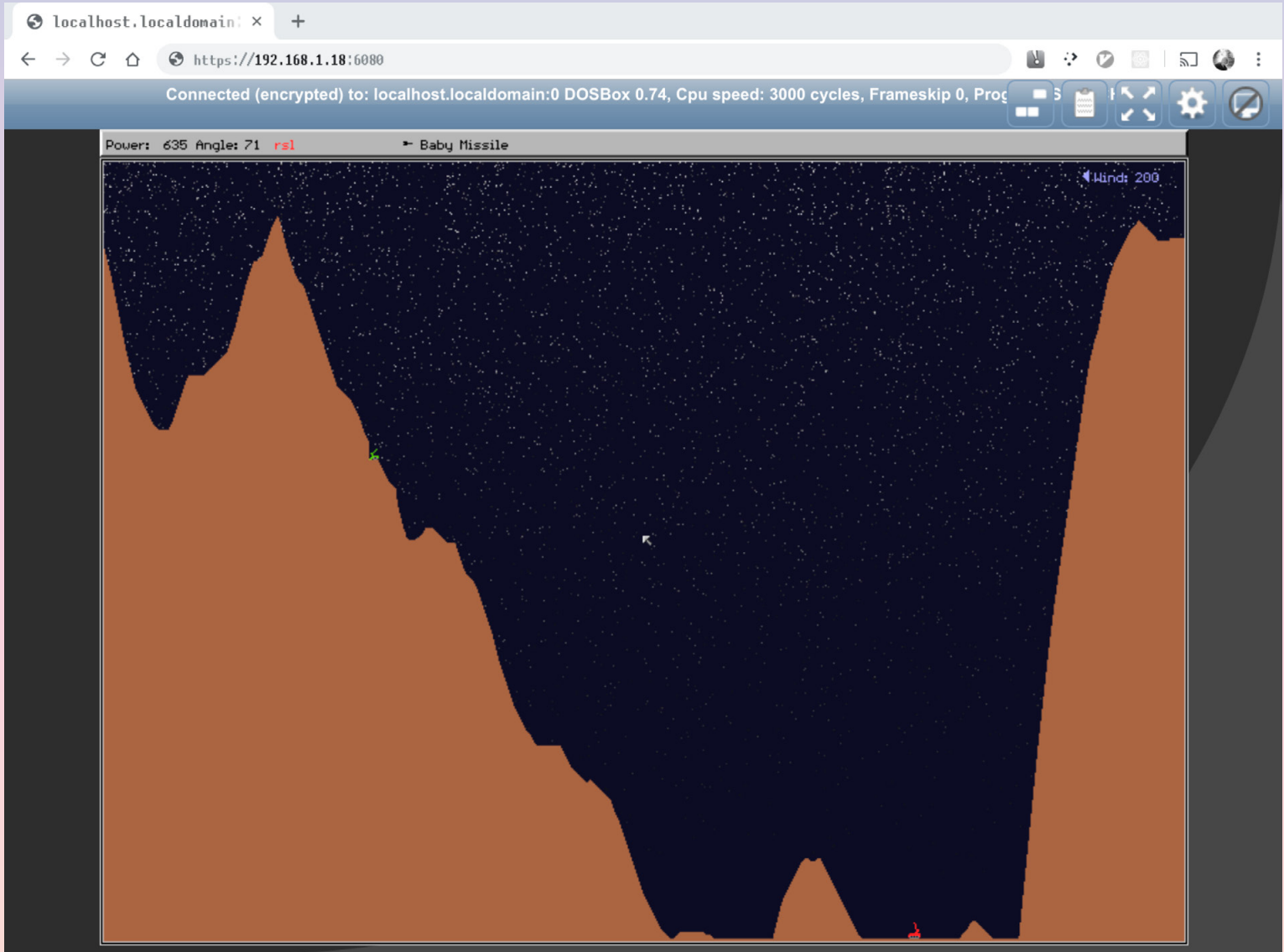
Oletuksena NoVNC käyttää porttia 6080. Se on kuitenkin mahdollista vaihtaa parametrilla `--listen PORT`, missä PORT tilalle määritetään haluttu portti. Tässä esimerkissä käytämme porttia 6080, mihin liikennöinti ei ole oletuksena sallittu palomuurista. Palomuuria vaivataan ja saadaan käyttöön seuraavilla komennoilla:

```
firewall-cmd --permanent --add-
port=6080/tcp
firewall-cmd --reload
```

Kun konfiguraatio on todettu toimivaksi, kannattaa luoda skripti, joka suorittaa toimenpiteet jatkossa:

```
#!/bin/sh
# Käynnistetään peli
dosbox ~/scorch/SCORCH.EXE &
# Odotetaan hetki (2 sekuntia),
että DOSBox käynnistyy
sleep 2
# Luetaan peliruudun id
muuttujaan $windowid
windowid=`xwininfo -root -tree |
grep DOSBox | awk '{ print $1 }'`
# Käynnistetään VNC-palvelin
x11vnc -sid $windowid -shared
-loop -forever -rfbauth ~/.vnc/
passwd &
# Käynnistetään NoVNC-palvelin
novnc_server --vnc 127.0.0.1:5900
--cert ~/cert.pem
```

Tallennetaan skripti esimerkiksi nimellä `tykkipeli.sh` käyttäjän kotihakemistoon. Tämän jälkeen määritetään sille ajo-oikeudet:



Scorched Earth -matsi käynnissä NoVNC:n yli.

```
chmod 744 ~/tykkipeli.sh
```

Nyt peli sekä VNC- ja NoVNC-palvelimet lähtevät käyntiin jatkossa vain ajamalla luotu skripti. Esimerkissämme käytössä on työpöytäympäristö LXDE, johon on myös mahdollista määrittää skripti ajettavaksi kirjautumisen yhteydessä. Tämä onnistuu lisäämällä seuraava rivi tiedoston `~/config/lxsession/LXDE/autostart` loppuun:

```
@/home/tykki/tykkipeli.sh
```

Samassa yhteydessä kannattaa näytönsäästäjä määrittää pois käytöstä kommentoimalla rivi `@xscreensaver -no-splash`:

```
# @xscreensaver -no-splash
```

Todellinen tykkipeli

Tässä vaiheessa peliin pitäisi päästä jo liittymään sisäverkosta yhdistämällä selaimella osoitteeseen `https://<sisäverkon osoite>:6080`, missä `<sisäverkon osoite>` on virtuaaliko-

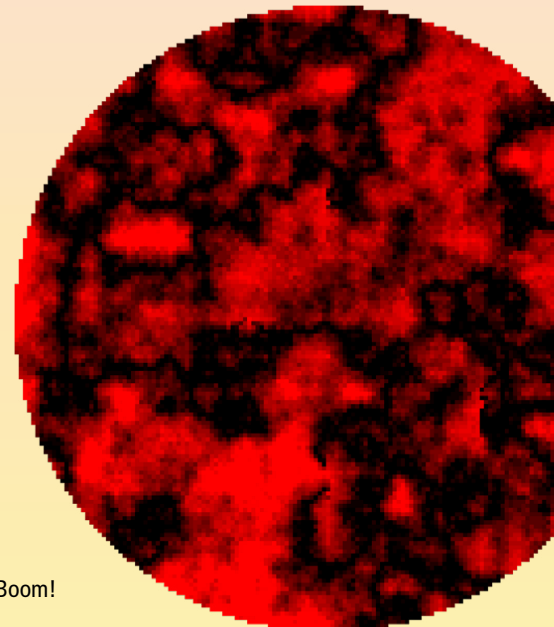
neen IP-osoite tai DNS-nimi. Peliin liittyminen Internetistä edellyttää vielä niin sanotun porttiohjauksen (port forwarding) luomista liittymän päätelaitteelle.

Porttiohjauksessa laitteelle kerrotaan portti, josta laite kuuntelee Internetistä tulevaa liikennettä, sekä osoite ja portti sisäverkossa, mihin Internetistä tuleva liikenne ohjataan. Asetuksen määrittäminen tapahtuu hieman eritavalla käytössä olevasta päätelaitteesta riippuen, joten yksityiskohtaisemmat ohjeet kannattaa tarkistaa laitteen manuaalista.

Porttiohjauksen läpi kulkeva liikenne on salattua http-liikennettä, jonka portti on Internet-standardissa 443. Tässä tapauksessa on järkevämpää käyttää jotain muuta porttia, esimerkiksi aikaisemminkin käytettyä 6080, ja sitä kautta vähentää todennäköisyyttä palveluun kohdistuvista ei-toivotuista yhteysyrityksistä. Määritetään siis porttiohjaukseen kuunneltavaksi portiksi 6080. Siihen tuleville yhteyksille kerrotaan kohteeksi NoVNC-palveli-

men sisäverkon osoite ja portti 6080.

Jos kaikki meni niin kuin pitää, peliä pitäisi päästä Internetistä pelaamaan kirjoittamalla selaimen osoiteriville `https://<ulkoverkon osoite>:6080`, jossa `<ulkoverkon osoite>` on korvattu liittymäsi IP-osoitteella tai siihen osoittavalla nimellä. 🚀



Boom!

**Aiheuttaako
IT-ongelmat
päänsärkyä?**



IT-PÄÄNSÄRKYYN

Tilapäiseen, toistuvaan tai krooniseen IT-päänsärkyyn.
Soveltuu kaikenkokoisille yrityksille toimialasta riippumatta.
Akuutissa tapauksessa soita 020 759 7541.

capnova.fi

1 980-luku oli siitä hienoa aikaa, että teknologia oli mainiossa kehitymisvaiheessa. Vaikka kirjoitin ekoja juttuja paperilla ja kirjoituskoneella, pääsin hyödyntämään tietokoneen mahdollisuuksia jo varhaisessa vaiheessa. Iso läpimurto oli oikea tekstinkäsittelyohjelma Commodore 64:lle. Se oli varmaankin alkujaan peräisin jonkin jenkkihlehden BASIC-listauksesta, mutta joku suomalainen – ehkä paikallinen hakkerigurumme **Kasper** – muokkasi siihen myös skandinavisen merkistön. Johan kelpasi kirjoittaa.

Ihan pikkuinen haaste tosin vielä oli: kuusnelosella ääkköset näkyivät ruudulla, mutta tulostuivat paperille aivan suttuna. Se oli pikku ongelma, kun jutut piti lähettää lehdistä paperille tulostettuna, jos ei halunnut hakata niitä kirjoituskoneella. Niinpä opetelin kirjoittamaan skandien sijaan erikoismerkeillä, jotka olivat ruudulla suttua mutta tulostuivat paperille oikein. Erikoismerkkien kirjoittaminen ja lukeminen sujuikin vuosikausia selköydintänsä. Myöhemmin siirtyminen Amigan *Wordperfectiin* oli iso loikka, kun piti opetella kirjoittamaan tavallisilla skandeilla ja ääkköset olivat näppikselläkin niin sanotusti tavallisilla paikoilla. Erikoismerkit tosin tekivät myöhemmin BBS-aikakaudella paluun, kun eri koneiden merkistöt näkyivät mikä minkäkinlaisena sekasotkuna (ks. Skrolli 2015.1).

Printtimedian kultakausi

Aika oli erilaista siksikin, että nykyään on äärimmäisen helppoa julkaista juttujaan blogeissa tai somepalveluissa. Samalla Instagramit ja Youtubet tarjoavat omanlaisensa kanavat sekä oikeasti luoville persoonille että turhakeille. Aivoilujaan saa ulos helposti, mutta itsensä monetisointi on hirvittävän vaikeaa. Etenkin verkkomediat toimivat niin pienillä budjeteilla, että niistä valtaosa maksaa vain säälipennejä, jos sitäkään. Moni pelisaiitti toimii pakon edessä yksinkertaisella vaihtosysteemillä: kirjoittaja saa arvostelukappaleen juttua vastaan.

Noilla vuosikymmenillä, aikana ennen internetiä, päivälehdet, sanomalehdet ja aikakauslehdet sen sijaan maksoivat nykypäivään verrattuna erittäin hyviä juttupalkkioita. Äitini muistelee, ettei poika ollut rahaa vailla kirjoittajauran alkamisen jälkeen. Sitä ennen



JOKSTORIAA

osa X: öääÖÄÄ{|}

Teksti ja kuvat: Jukka O. Kauppinen

Edellisessä JOKstoriassa pääsin viimein asiaan eli lehtikirjoittamisen alkeisiin. Jatketaanpa samalla linjalla, aloittelevan toimittajan muistelmissa.

olin tienannut taskurahoja muun muassa mainoksia jakamalla, eli lienen pärjännyt varsin omillani jo varhaisteinikäisestä lähtien.

Kirjoittamisesta oli nuorelle jolpille muutakin kuin rahallista iloa. Aloitin peliarvosteluista, mutta siitä oli lyhyt askel kirjoittamaan myös uusista sarjakuvista, kirjoista ja musiikista. Oli muuten

aivan mahtavaa, kun pystyin kävelemään musiikki- tai kirjakauppaan ja ostamaan mitä vain uusia julkaisuja. Sitten vain tein jutut ja kärräsin tekstit toimitukseen kuittien kera. Lehti korvasi ostokset, maksoi juttupalkkion, ja minulla oli hyllyssä taas jotain uutta, ehkä kivaakin.

Muun muassa levyjä tuli hankittua laajalla skaalalla, koska täytyihän

Mikromania

Tähtivaeltaja: suomalainen sci-fi-lehti

Suomalainen science fiction on jäänyt vailla arvostusta, kuten sci-fi yleensäkin. Vieläkin törmää hölmöihin, jotka eivät tajua sci-fista mitään ja siksi halveksuvat sitä.

Mutta... valopilkkuakin löytyy. On olemassa tähtivaeltaja, lehti joka on suomalaisen sci-fin edelläkävijä. Tähtivaeltaja toimii sci-fin sanansaattajana meille raukoille. Yleisesti on tunnettu TV:n olevan paras suomalainen sci-fi-lehti, jonka ennakkoilto on rohkea linja on saanut satoja kannattajia.

Tähtivaeltajaa julkaisee Helsingin Sci-Fi-seura, ja päätoimittajana & puhtaaksikirjoittajana & laittajana toimii itse Toni Jerro, ympäri Suomenmaan tunnettu sci-fi-fani. Hänestä voitisi kertoilla juttua paljonkin, mutta antaa olla.

Vastikään ilmestyi TV 2/87. Ensimmäinen numero ilmestyi kesäkuussa, ja Toni yritti selittää synnä ollen suunnaton vastoinkäymisten summa, mutta meidän emme tuota usko. Todennäköisesti ukko on maannut pöydän alla krappulan kou-

Jerry-miehen sarjakuvien SF-fanien ruozinmatkan kertomus (Kuitenkin Me Kuulemme Eikä Elämäsi Niin Kuin Me — nimesä kerakseen). Näyttää olevan ilmiö, että yleensäkin paras kaikissa SF-lehdissä ovat con yms. kertomukset. Tyyppi hilumassa ruozissa kotibileissä ja poistuessa yksi otus hokee "we can't go out, it's a sweden out there". Niin sitä pitää.

Reaktiot TV:n ovat olleet yleensä kommentteja tyylisiin "täysin sairasta". Juuri niin. Mitä sairaampaa sen parempaa. Se olkoon Tähtivaeltajan tuleva linja.

Mikäli kiinnostusta lehteä kohtaan löytyy voi tilata näyttenumeron. Numeron hinta on 15 markkaa. Näyttenumeroksi kannattaa ottaa nro 4/86, 1/87 on Philip K. Dick specialia. Eikä anna ehkä oikein kunnollista kuvaa lehdestä. Vuosikerran tilaus hinta on 50 markkaa ja valuiltua PS-tilille 3143 67-6. Saajana Helsingin Science Fiction Seura, Viihdejantte 4-6 D 103, 00410 Helsinki. Numerot tulenevat lähiaikoina.

Mutta parasta oli jälleen

...riissa, vaikka on se saattanut pyöriä Backstagellakin. Mutta oli miten oli. Uusi numero ilmestyi silti. Vuorossa oli tällä kertaa Sarjakuvu-specialia.

TV 2/87:ssä esitellään kovasti suomalaisia sarjakuvapiirtäjiä, ja mukana on tietysti heidän sarjiksiinkin, kuten Markku Uusitalon 'Kapteeni Hyperventilaattorimies Pelastaa Joulu'. Kyseinen sarja nousi välittömästi suosikkisarjoihin joukkoon. Hirvittävät naurukohtaukset ravistelivat maailista tommujajani lukiesani sitä. Jerry-miehiä, jotka muistuttavat epäilyttävästi sekä nimeltään että ulkonäkönsä, puhumattakaan tavoistaan. Todella loistavaa!

Mukana on myös muitakin hienoja sarjakuvia, kuten Kiven 'Vanilium Vanitas Et Omne Vanitas'. Myös hyvä Alfred Besterin käännösnovelli 'Hellät Fahrenheitit' on mukana. Myönteään että en ole kuullut ukosta ennen, mutta novelli kolhoi räjästi päähän. Liisä hänen tuotantoani!

Kuten edellisessä JOKstoriassa paljastin, kirjoitin myös salanimillä...

82 SKROLLI 2019.3

juttuja tehdä muistakin kuin omista lempijutuista tai aiheista, joita tunsin etukäteen. Se toki johti yllätyksiin, kun vastaan tuli outoja bändejä, joiden musiikkityylissä riitti päänraapimista. Aikana ennen internetiä tiedonhankinta oudoista aiheista oli erittäin rajallista, joten juttu täytyi kirjoittaa korvakuvolta, hyvinkin hatarilla pohjatiedoilla. Mutta ihan kelpo tarinoita ne taisivat olla, kun kerran kelpasivat. Koska kirjoittaminen tuntui sujuvan, aiheiden skaala kasvoi nopeasti. Pienet tietokonepalstat ja yksittäiset pikkujutut kasvoivat nopeasti suuremmiksi lukujuutuiksi ja juttusarjoiksi.

Kekseliäs veijari kehitteli myös omia intressejä edesauttavia juonia. Koska olin ihastunut avaruusjuttuihin jo pienenä, siitä kelpasi kasvaa science fictionin suuntaan. Jotenkin törmäsin *Tähtivaeltaja*-lehteen, ja sen kautta Ylä-Savon ytimeen avautui ulottuvuusportaali, joka kertoi neljä kertaa vuodessa kaukaisesta aikuisten scififanien maailmasta, jossa seikkailtiin koti- ja ulkomailla ja tehtiin kaikenlaista outoa. Lehteä lukemalla opin etsimään kirjastosta aivan uudenlaisia, kummallisia levyjä, kirjoja ja sarjiksia, joihin en olisi muutoin koskaan tarttunut. Tutustuin uusiin elokuvagenreihin ja sukelsin esimerkiksi kauhuelokuvien uumeniin lukemalla niistä – leffojahan ei meilläpäin juuri kiertänyt. Suomalaisesta scifi-fandomista lukeminen avasi maailmankuvaani tehokkaasti, ja haluankin nostaa hattua suomalaisen scifin väsymättömälle puurtajalle **Toni Jerrmannille**. Hänen vallattomat, lonkerohuuruiset tarinansa kummallisine musiikkisuosituksineen olivat kuin toisesta maailmankaikkeudesta.

Yksi nerokkaista leimauksistani olikin tehdä parikin eri juttusarjaa suomalaisista pienlehdistä ja tietenkin science fiction- ja sarjakuvakerhojen julkaisuista. Se oli näet loistava tekosyy tilata kaikki mahdolliset lehdet ja vielä saada rahaa niiden lukemisesta. *Tähtivaeltajan* lomassa tutuksi tulivat *Portti*, *Spin*, *Aikakone*, *Sarjainfo*, *Suuri Kurpitsa* ja monet muut pienlehdet.

Arvostelukappaleita puhelinskillseillä

Tein tietysti koko ajan juttuja peleistä ja tietokoneiden hyötyohjelmista. Mutta mistä hommasin jutunaiheeni? Edes kylän turbotape-kingi ei pysy ajan hermolla pelkillä piraattipeleillä.

Ratkaisu oli kaksi- tai kolme- osainen: ensin taruttiin puhelimeen. Kun kerran kirjoitin säännöllisesti lehtiin, niin se riitti meriittiksi maahantuoille ja edustajille, jotka lähettivät tuoteuutuksiaan juttuja varten. En muista, miten pelit valittiin. Mahtoivatko he suosittelua uutuutuksiin puhelimesta vai katselinko lehtien ilmoituksia? Varmaan sekä että. Ilmaista herkkuaahan nämä arvostelukappaleet eivät olleet, sillä muistaakseni pelit olivat vain lainassa. Elimme kuitenkin siitä erikoisia aikoja, että Posti palveli hyvin, nopeasti ja jopa edullisesti. Isompienkin pelipakettien lähettäminen takaisin maahantuojalle mah- tui hyvin pikku kirjoittajan budjetteihin.

Toinen tapa saada arvostelukappaleita oli luoda hyvät yhteydet paikallisiin kauppiaisiin, joita meillä päin olivat kaupakettjut, kodinkoneliikkeet ja etenkin Info-kirjakaupakettjun tietokonepuoli. Ei ehkä ollut kummoinen yllätys, että tiesin usein enemmän peleistä kuin keskiverto jääkaappi- ja stereomyyjä, joten välillä päädyin myös neuvonantajaksi. Mitä pelejä kannattaa tilata myyntiin ja mitä ei? En nyt sano, että olisin ollut aina oikeassa, mutta ainakin kauppiat osasivat olla tilaamatta vanhinta kuraa hyllyilleen.

Infossa oli muuten erinomaisen asiantunteva myyjä, joka oli aktiivinen tietokoneharrastaja. Hänellä oli näköiseni pelimaku. Vaihtelimme strategiapeljä ja tulimme muutenkin hyvin juttuun, mistä oli kerran yllättävää

Jukka O. Kauppinen alias Grendel/Byterapers on pitkän linjan toimittaja, joka kirjoitti ensimmäisiä tekstejään mekaanisella kirjoituskoneella. JOKin ensimmäinen peliarvostelu julkaistiin 1986, mistä lähtien hän on ahkeroinut tauotta kirjoittaen juttuja muun muassa videopeleistä, tietotekniikasta, viihteestä, ilmailusta ja burleskista. JOKistoriaa-sarjassa pureskellaan niin kotimikroilun kuin peli- ja digijournalismin maailmaa grennulasien läpi nähtynä. jukka@skrolli.fi

10 — Perjantaina 24. huhtikuuta **Mikromania** **SALMETAR**

Mikromania

Gunship-Microprose
Ja tässä on kaikille lentämisestä ja kovasta sotimisesta piäläville: Gunship! Maailman tehokkain taisteluhelikopteri AH-64 Apache on nyt saatavissa. Suuria odotuksia ei petetty. Hieno vektorigrafikka, todellista lentämisen tuntua... kaikkea. Pakkiaan ei puutu.
Taistelun on käytettävissä harjoitusalue USA:ssa ja neljä sotaskenariota Aasiasta Länsi-Eurooppaan. Apachesi on aseistettu 30 mm tykillä, AGM-114 panssarintorjuntajohkeilla, 2.75" FFAR raketeilla ja AIM-9L ohjuksilla, joskin voit valita mitä aseita voit lastata. Mukana on tutka- ja infrapunalaitteiden sekoittimet. Vihollisella on käytössä lähes kaikkia muita joutokkoja paitsi polkupyöräsukseltajia. Taistelussa onkin tärkeää osata erottaa omat yksiköt vihollisista, ettei tule vahinkoa.
Grafikka on erinomaista vektorigrafikkaa. Maastossa on kukkuloita, joiden lomasta hyökätessään voi aiheuttaa monille vihollisille vetelät housuun (jos he enää siihen pystyvät). Pelin ääninähdykset ovat lähinnä vain moottorien ulinaa, tulituksen ja vihollisen osumia, mutta kuika muuta kaipaisikaan.
Hemmetin hyvä peli!
Gra 9, ään 8, pel 9, kii 9 ja kokonais 9.

Raid 2000 -Mirrorsoft
Pelin ideana on pelastaa maapallo muukalaisten hyökkäykseltä. Eläköön kaikki uudet ideat! Käytettävissä on neljä tehokasta hävittäjää, joiden aseistuksena on kaksi tykkiä ja yksi ydinpommi.

Championship Wrestling -U.S. Gold
Tämä "Rock 'n' Wrestling" manteliperijä sai aikaan melkoisia odotuksia, joita se ei kuitenkaan kaikkia täytä. Kasettiversiosta puuttuu pelin paras osa, ottelijoiden "kommentit", ja tietenkin la- taukset ovat hitaanpuoleiset.
Valittavissa on kahdeksan eri painijaa, joilla kullakin on oma erikoisotteensa. Liikkeitä on valittavissa enemmän kuin tarpeeksi, ja tuskinpa monikaan koskaan niitä "kaikkia" oppi- kaan. Pelattavuudella ei voi paljoakaan kehua, monet liikkeet kun vaativat melko monimutkaisia väänönsiä tikulla.
Ukkelit tuntuvat aluksi turhan pieniltä, mutta ajan mittaan huomaa, ettei koko haittaa. Onpahan isompi pelialue.
Gra 9, ään 8, pel 8, kii 8 ja kokonaisuus 8.



Ruutukaappausten taltiointi ei ollut siihen aikaan aivan yhtä käte- vää kuin nykypäivänä, joten peliarvioiden kuvituksena oli tässäkin artikkelissa vallon jotain muuta.

hyötyä: hän näet päivänä eräänä ker- toi, että myymälässä oli käynyt ihan oikea amigisti Pielavedeltä. Meitä ei silloin montaa ollut, joten hän antoi minulle kaverin puhelinnumeron, ja näin sain hassulla tapaa yhteyden pelivelveläiseen tornitalon mittaiseen kaveriin ja hänen veljiinsä ja ystäviin- sä, jotka olivat kaikki kovia bitinnäp- läjää. Aikanaan näistä tyypeistä tuli **Proton**, **Nico**, **Lasse**, **Dr. Dick** ja **Mr. Sex** of Byterapers. Voi sanoa, että suo- malainen demoskene on tälle Infon myyjälle paljon velkaa.

Seuraavassa JOKistoriassa lisää mui- naista peliarvostelujorinaa ja ehkä vä- hän valokuvaustakin. 📷

SKROLLI 6,4 VUOTTA SITTEN

Terve-
tuloa
Skrolli
6,4 vuotta sit-
ten -palstan
toiseen osaan,
jossa jatkamme
aikamatkaa

Skrolli-lehden historiaan.

Edellisessä osassa tutustuimme lehden monistekokoiseen maistiaisnumeroon 2012.0 ja nyt on tutkailussa kaikkien aikojen ensimmäinen varsinainen Skrolli. Olipa kerran synkkä ja myrskyisä yö, tarkalleen 6,4 vuotta sitten...

Huhtikuu 2013 oli Skrolli-historian merkkipaalu: edellisen kuun lopulla julkaistu Skrolli-ensinumero 2013.1 myytiin loppuun 10.4., vaikka lehti oli julkaistu vasta paria viikkoa aikaisemmin 25.3. ja pdf-versiokin oli laitettu makсутta kaikkien ladattavaksi jo 4.4. Osviittaa antaneesta ennakkomyynnistä huolimatta paperisen tietokonelehden kysyntä yllätti toimituksen. Viimeistään nyt oli selvää, että Skrollia oli kaivattu. Kerroimme historian tarkemmin numerossa 2015.1.

Ihan vielä Skrolli ei kuitenkaan sauttanut vakiintunutta muotoaan – kirjaimellisesti. Pokkarimaisen liimaselän sijaan Skrollin ensimmäinen vuosikerta nidottiin stiftaamalla eli vihkomaisesti parilla niitillä. Yksi syy eroon on varhaisten lehtien pienempi sivumäärä. Siinä missä kirjoitushetken uusin Skrolli 2019.2 kansineen on pituudeltaan 92 sivua, Skrolli 2013.1:n mitta oli 52 sivua.

Nidontaeron huomaa pdf-versioissakin, vaikka aikakauslehdelle poikkeuksellinen A4-koko oli käytössä alusta alkaen: Stiftattujen Skrollien kannen laidassa ei ole selkämysmerkintöjä, koska lehdillä ei ole selkämystä. Tosin ensimmäisestä liimasidotusta vuosikerrasta 2014 varsinaiset selkämysmerkinnät puuttuvat myös, taittajan mukaan unohduksen vuoksi.

Suuri osa numerosta 2013.1 oli taitettu 2012.0:n tapaan kahteen palstaan, joskin paikoitellen kokeiltiin kolmea, jopa neljää palstaa. Kaksipalstainen taitto vaipui 2013.1:n myötä pitkälti historiaan, mutta kolme ja neljä palstaa kilvoittelisivat sivuilla vielä vuoteen 2014, jolloin nykyinen kolmen palstan taitto vakiintui. Tarkka silmä huomaa eroja myöhempään myös kirjasintyyliessä.

Sisällön askelmerkit olivat kuitenkin jo tuttuja alusta alkaen. Olkaapa hyvät, kolme otetta Skrollista 2013.1:

"Skrolli-lehti on syntynyt maailmaan, jossa tietokoneet ovat etupäässä kulutushyödykkeitä. Kaikki käyttävät niitä, mutta harvempaa kiinnostaa ymmärtää niitä. Jopa suomalaiset tietokonelehdet ovat nykyään lähinnä osto-oppaita, joista inhimillinen tekeminen ja innostus on syytä pois kulutuskeskeisyyden tieltä. Aikoinaan nämä lehdet inspiroivat kokonaista uutta osaajasukupolvea, mutta nykyään eivät osajiksi haluavat löydä niistä enää mitään kiinnostavaa."

– **Ville-Matias Heikkilä**, Pääkirjoitus: Kokonaiskuva ja jatkuvuutta

"Siellä vuodessa 2030 olette jo päässeet kokemaan sen, kun maailmaan [3D-]tulostettiin ensimmäiset aseet, räjähteet ja huumeet. Olette kokeneet sen paskamyrskyn, joka syntyi, kun Monsanto patentoi tuja genejä printattiin synteettiseksi siemeneksi. Tiedätte, että tekijänoikeuksien vahtikoirat ja sensuurihimoiset uraohjuspoliitit olivat vain ensimmäinen askel vanhan vallan ja korporaatioiden johtamassa sodassa tietokoneita ja erityisesti tietoverkkojen mahdollisuuksia vastaan." – **Tapio Berschewsky**, Kolumni: Viestejä ylihuomisen lukijalle

"Konkurssin yhteydessä [Aтарin pelikonsolin] Jaguarin kuorimuotit myytiin yritykselle, joka prässäsi niillä hammaslääkärin hoitolaitteita. Sellaisella suoritettu toimenpide olikin varmasti useimmille se miellyttävän ja



Lisäys musalistalle

Kiitoksia viimeisimmästä Skrollin numerosta 2019.2. Silmäilin läpi mielenkiintoisen jutun musasoftista. Listalta puuttui entisen Cakewalk Sonarin seuraaja, Bandlab, joka on nykyään saatavissa ilmaiseksi. Kyseessä on siis täysi ammattitason softa, joka pohjautuu Sonarin viimeiseen Platinum-versioon ja jota kehitetään edelleen. Mukana tulee hurjasti virtuaali-instrumentteja yms., joten paketti on loistava kelle tahansa musiikintekijälle.

Itselle aikanaan kalliin musasoftan ostamiseen oli melkoinen kynns. Tuolla pääsee ilmaiseksi kokeilemaan, miltä ammattisofta tuntuu ja miten sellainen toimii. Softa löytyy osoitteesta www.bandlab.com/products/cakewalk ja on sieltä ladattavissa rekisteröintiä vastaan.

Aki-Ville Auvinen

Kiitos hyvästä vinkistä! Artikkelisi musiikinteko-ohjelmista sai kiitosta ja herätti keskustelua – kaikilla tuntui olevan mielipide listasta. Toinen ohjelma, jota kaipahtiin, oli entinen Linux MultiMedia Studio, nykyinen LMMS (lms.io). Kyseessä on avoimen lähdekoodin DAW, joka toimii useilla alustoilla.

Omasta mielestäni suurin puute oli tietenkin OctaMED (sitten MED Soundstudio), mutta myönnettäköön vastentahtoisesti, että trackerit kuuluvat omaan artikkeliinsa. :) Se artikkeli oli muuten Skrollissa 2014.3 (pdf-lehti maksutta: skrolli.fi/numerot).

Janne Sirén

antoinen kokemus Jaguarin parissa." – **Mikko Heinonen**, Ei näin: Atari Jaguar, kissapedon kuolonkorinat

Lehdessä oli myös yksi palsta, joka ei ole toistunut sittemmin: Seuraavan Skrollin aiheita. Kävi nimittäin hyvin nopeasti selväksi, kuinka vaikea talkoo-projektin juttuputkea on ennustaa... 🐼

Janne Sirén

Skrollin vanhat pdf-lehdet saat maksutta osoitteesta: skrolli.fi/numerot



SKROLLIKALENTERI

Tervetuloa neljänteen Skrolli-kalenteriin – kalenteri tekee täyden ympyrän, kun kokonainen vuosi suomalaisittain kiinnostavia tapahtumia on nyt listattu! Skrolli-kalenterissa kartoitamme tietokonekulttuurin kokonaiskuvaa päivämäärien ja tapahtumapaikkojen kautta. Alla kattaus Suomessa ja suomalaisittain kiinnostavissa maailmankolkissa järjestettäviä tietokone-, peli- ja demoskenetapahtumia sekä huomionarvoisia yhteisötapahtumia.

PVM	TAPAHTUMA
Kulttuuri & yhteisöt	
28.9.2019	Saku 2019, Oulu. Suomen Amiga-käyttäjät ry:n perinteisessä kaikille avoimessa ja ilmaisessa tapahtumassa pääsee pelaamaan vapaasti Amiga-tietokoneilla, katselemaan Amiga-demoja, voittamaan palkintoja pelikilpailuissa, sekä tutustumaan uusiin laitteisiin ja ohjelmistoihin. AMIGAAA!
28.9.2019	David Lynch -elokuvien musiikkia, Tampere-talo. Gasthaus Orkesteri soittaa kuuluisan ohjaajan David Lynchin elokuvia, kuten Mulholland Drive, Eraserhead ja Lost Highway musiikkia. Video- ja valoshow täydentää tunnelman.
29.9.2019	Animeseminaari, Oulu, Kulttuuritalo Valve. Teemana Isekai – matka toiseen maailmaan, jonka innoittamia ohjelmia, pajoja, kisoja sekä toimintaa on luvassa koko päivän mitalta kaikenikäisille. Tapahtuma on maksuton kaikille kävijöille!
12.–13.10.2019	Amiga34, Rheinische Landestheater, Neuss, Saksa. Amiga-kotitietokoneen 34 v. -synttärikemut naamioituna tietokonetapahtumaksi.
9.11.2019	Kitsunecon, Kaariina-talo. Tapahtuma esittelee japanilaiseen populaarikulttuuriin liittyviä ilmiöitä ja harrastuksia, kuten mangaa, animea ja cosplayta. Kaarina-talossa järjestettävä con on maksuton ja sopii kaikenikäisille.
23.–24.11.2019	Tracon Hitpoint, Tampere Tredu. Pöytäpelaamiseen keskittyvä tapahtuma tarjoaa lauta-, kortti-, rooli- ja miniatyyripelejä sekä muuta aiheeseen liittyvää ohjelmaa. Tapahtuma sopii kaikille pelaajille ensikertalaisista kokeneisiin konkareihin. Hitpoint kestää koko viikonlopun.
Demoskene	
28.9.2019	Skrolli Party 2019, Tampere. Järjestyksessään kolmas Skrolli-lehden järjestämä demoparty – tällä kertaa Espoon sijaan Tampereella.
4.–6.10.2019	ZOO'19, Viialan Seurahuone, Akaa. Suomen suurin Commodore 64 -tapahtuma. Paikalta löytyy musiikkia, demoja, juomaa, sauna ja tajunnan rajat avaava 8-bittinen kokemus. Digitaalisen taiteen kilpailuja kaikissa mahdollisissa luokissa sekä runsas määrä live-esiintyjä ja seminaareja.
Yhteisölliset työtilat	
Helsinki Hacklab	avoimet ovet tiistaisin, kurssipäivät torstaisin
Tampere Hacklab	avoimet ovet tiistaisin, workshopit torstaisin
Vaasa Hacklab	avoimet ovet keskiviikkoisin, kurssit ja workshopit tiistaisin
Turku Hacklab	avoimet ovet tiistaisin, kuun 1. tiistai show & tell -ilta
Kouvola Hacklab	avoimet ovet tiistaisin
Nokia Hacklab	avoimet ovet torstaisin
Oulu Tarlab	avoimet ovet torstaisin
Hacklab Jyväskylä	avoimet ovet tiistaisin, kerhoilta lauantaisin
Hacklab Mikkeli	avoimet ovet torstaisin
Hacklab Lahti	avoimet ovet keskiviikkoisin
TechClub Joensuu	avoimet ovet keskiviikkoisin

Hacklabit ja yleisemmin yhteisölliset työtilat tai hackerspacet ovat paikkoja, joissa voi rakentaa ja toteuttaa kiinnostuksensa kohteena olevia asioita. Ne ovat yhteisön ylläpitämiä verstaiteja. Niistä voi löytää esimerkiksi elektroniikan työpisteet, metallipajan, puuverstaan tai luokahuoneen koulutuksiin sekä välineitä, kuten 3D-tulostimia, oskilloskooppeja tai vaikka ompelukoneita.

Tarkista osoitteesta hacklab.fi ajan tasalla oleva listaus Suomen hacklabista. Samalta sivulta löydät myös linkit kunkin hacklabin sivuille, joista voit tarkistaa avoimien ovien ajankohdat ja saapumishjeet.

PVM	TAPAHTUMA
Esports & lanit (Suluissa paikkamäärä)	
4.–6.10.2019	Salo Gaming Expo, Salohalli, Salo (390)
11.–13.10.2019	QrikkaLan, Kampus, Kurikka (500)
17.–19.10.2019	Grail Quest, Kupittaaan palloiluhalli, Turku (300)
17.–20.10.2019	Gyostage 2019, Kuopio-halli, Kuopio (736)
24.–27.10.2019	Insomnia XXI, Karhuhalli, Pori (774)
1.–3.11.2019	Skynet Langames 25, Holiday Club Saimaa, Lappeenranta (500)
8.–9.11.2019	Pelimiitti LAN syksy 2019, Raudaskylän opisto, Ylivieska (40)
Peliala	
18.9.2019	IGDA Turku Hub Gathering with Fakefish
24.9.2019	IGDA Kajaani Hub Gathering
30.9.2019	IGDA Leadership Day Helsinki
2.10.2019	IGDA Kuopio Hub Gamesviikko Gathering
8.10.2019	IGDA Helsinki Hub Gathering
15.10.2019	IGDA Turku Hub Gathering
6.11.2019	IGDA Kuopio Hub Gamesviikko Gathering
12.11.2019	IGDA Helsinki Hub Gathering
15.11.2019	W <3 Games Conference Helsinki
19.11.2019	IGDA Turku Hub Gathering
3.12.2019	IGDA Turku Hub Xmas Party
4.12.2019	IGDA Kuopio Hub Gamesviikko Gathering
10.12.2019	IGDA Helsinki Hub Gathering
Suomen Pelimuseo, Tampere	
15.9.2019	Pelataan: Harry Potter ja muut maagiset olennot, klo 12–17.
25.9.2019	Pelipiiri, klo 18–20.
27.9.2019	Astraterra-roolipeli, klo 15–18.
27.9.2019	Pelillisen valoteoksen syyttäminen pelimuseon ikkunoihin, klo 17.
12.10.2019	Pelataan: Pelejä Tampereelta, klo 12–17.
19.10.2019	Collaborative game histories -seminaarin yleisöpäivä. Tapahtumaan vapaa pääsy, klo 10-18.
23.10.2019	Pelipiiri, klo 18–20.
25.10.2019	Astraterra-roolipeli, klo 15–18.
3.11.2019	Museocon. Ohjelmassa esitelmää, opastettuja näyttelykierroksia sekä tietenkin pelejä läpi päivän, klo 12–17.
10.11.2019	Pelataan: Super Mar10, klo 12–17.
20.11.2019	Pelipiiri, klo 18–20.
28.11.2019	Pelienvaihtotori Vapriikin pikkujouluissa, klo 17–20.
29.11.2019	Astraterra-roolipeli, klo 15–18.



Tapahtuma-järjestäjä tai -kävijä, vinkkaa meille tapahtumastasi – myös pienempien paikkakuntien tilaisuuksien ovat tervetulleita: kalenteri@skrolli.fi

Quake SM 2019 ja Assembly Summer

Skrolli-lavalta jyrähti

Skrolli osallistui omalla ständillään Assembly Summer -tietokonefestivaaleille jo seitsemättä kertaa. Nyt osastollemme nousi Skrolli-lava.

Teksti: Janne Sirén

Kuvat: Toni Kuokkanen, Tanja Kuokkanen

Elokuun ensimmäisenä viikonloppuna tuttu ilmoitus raikui Helsingin messukeskuksessa: ”Kill all audio and lights!” Skrolli-lavalta kaikui kuitenkin toinen Assy-klassikko, jota 2010-luvulla ei ole niin usein kuultu: ”Quaket v#&e%&n!” – eli huuto, jolla joskus silloin lähempänä demoskenen kulta-aikoja Assembly-yleisö pyrki vaientamaan ne pahuksen pelaajat demokompojen tieltä.

Skrolli ei kuitenkaan muistellut legendaarista huutoa koodauskisojen tähden, vaan käänsi historiallisen asetelman päälaelleen: huudosta oli tullut Skrollin Assemblyilla järjestettävän Quake SM -turnauksen epävirallinen tunnuslause. Jotta itseironinen pyllistys historialle olisi tahattomasti täydellinen, Skrolli-lava oli plaseerattu tapahtuman Scene Lounge -tilan viereen. Quake-huudatuksemme kaikui komeasti (puolustukseemme: myöhässä alkaneen) 4k-introkompon päälle.

Kiitos ja anteeksi.

Pelejä ja puhetta

Elektronisen urheilun liitto SEUL oli tukemassa Quake-turnaustamme, ja puheenjohtaja **Joonas Kapiainen** vieraili lavalla jakamassa pääpalkinnon – kiitos! Quake-finaalin juonsi itseoituksellisesti Suomen e-urheiluselostajien ykköskanuuna **Teemu ”Wabbit” Hiilinen**. Skrolli-lava ei kuitenkaan ollut pelkkää Quakea. Lavalla vieraili kymmenkunta Skrollin kutsuvierasta, kuten legendaarisen Llamasoftin (*Attack of the Mutant Camels*) **Jeff ”Yak” Minter** ja **Ivan ”Giles” Zorzin**, jotka kertoivat Netflixin *Black Mirror: Bandersnatch* -sarjan teosta sekä pelinkehityksestä 1980-luvulla.

Myös **Petri Saarikosken** esitelmä BBS-kulttuurista herätti paljon kiinnostusta, ja nähtiinpä lavalla Skrollin omaakin väkeä. Esimerkiksi **Ei näin!** -palstan **Mikko Heinonen** jalkautui yleisön eteen kuin standup-koomikko konsanaan. Tässä olisi toden teolla

Skrolli-esittäjä: *Ei näin!* -kiertueen ainesta. Skrolli-lavan koko ohjelman voit palautella mieliin oheisesta laatikosta.

Tuttuun tapaan Skrollin osastolla oli myös maksuttomia kolikkopelejä. Pelit olivat lainassa Suomen Pelimuseon ja sen tukijoiden kokoelmasta, yhteistyössä kumppanimme Retro Rewind -pelilehden kanssa. Tämän vuoden pelit olivat *Missile Command*, *Turbo Outrun* ja *Xevious* sekä lisukkeena Arcade1Upin versio *Street Fighterista* – Assemblyjen timelapse-video muuten vahvistaa, että pelejä pelattiin kirjaimellisesti yötä päivää!

Skrollilla oli myös omaa kerrottavaa. Suuri uutisemme Assyilla oli, että Retro Rewind -lehti ilmestyy ensi vuonna Skrollin liitteenä. Innokkaimmat tilasivatkin 2020-vuosikertamme jo tapahtumassa. Myynti jatkuu verkossa: tilaaskrolli.fi.





Assemblyn pääjärjestäjä Pekka "Pehu" Aakko näyttämässä #SkrolliQuake-po-seerausmallia.

Onnea kuvakisan voittajalle!

Quake SM -turnauksen lisäksi Skrolli järjesti Assemblyllä #SkrolliQuake-kuvakilpailun, johon pääsi mukaan julkaisemalla sosiaalisessa mediassa aihe-tunnisteella varustetun kuvansa Skrolli-osaston pahvisen Quake Rangerin kanssa. Palkintona oli konepaikkalippu seuraavaan Assembly Winter -tapahtumaan. Juuri parahiksi lehden mennessä painoon voittajaksi arvottiin Jesse "Furyfare" Santaoja. Onnea!



Reiska palasi kotiin. LuGia voitti myös edellisen Skrolli <3 Quake -turnauksen vuosi sitten (ks. Skrolli 2018.3), joten Skrollin Ranger-kiertopokaali lähti takaisin samaan osoitteeseen. Kuvassa Quake SM:n top 3: Johannes (vas.), LuGia sekä jjuhuo. Oikealla SEULin Joonas.

Quake SM 2019 -tulokset

1. LuGia
2. jjuhuo
3. Johannes
4. Plokk

Skrolli-lavan ohjelma

Torstai

Quake SM – Karsinnat

Perjantai

Quake SM – Karsinnat

Iiro Uusitalo – Team Whack CASE: Fidelix

Jeff Minter -esitelmä #1 – Game development since the early 1980s

Quake SM – Bracket

Lauantai

Petri Saarikoski – Kokemuksia ja muistoja BBS-harrastuksen valtakaudelta

Mikko Heinonen – Ei näin! (Greatest Hits)

Jeff Minter -esitelmä #2 – Behind Black Mirror: Bandersnatch

Art of Coding: Demoscene Unescon aineettomaksi kulttuuriperinnöksi

– Paneelikeskustelu

Quake SM – Finaali

Hatunnosto Skrollin digipäällikkö **Toni Kuokkaselle**, joka organisoii koko Assembly-osanotomme, ja **Jukka O. Kauppiselle** tulituesta ohjelman rakentamisessa.

Me ja kumppanimme videoimme osan peleistä ja esityksistä. Jos kiinnostuit, kytää Skrollin sosiaalista mediaa ja osoitetta skrolli.fi/assembly.



Unohtumattoman unohdetut demokompot



Teksti: Jukka O. Kauppinen Kuvat: Janeway, Pouët.net, kirjoittajan lähteet

Vanha ei aina jaksa, mutta ei kyllä nuorempikaan. Assemblyjen tämänvuotisen demoskaban ajastaminen alkuiltaan oli upea päätös, joka herätti myös muistoja niistä muinaisista demoskaboista, joissa aikataulut eivät pitäneet kutiaan.

Olen vuosikymmenten mitaan käynyt erinäisillä kymmenillä demopartyillä, joskin viime vuosina vain laiskasti. Taipaleen varrelle mahtuu monia demokisoja, joista osassa kompoihin osallistui myös oman ryhmämme Byterapersin teoksia. Vaikka olenkin kaikin puolin kädetön ja taidoton tekemään mitään järkevää, niin olen osallistunut muutaman kerran kisoihin itsekkin, eikä minua ole edes aina pudotettu alkukarsinnassa! Hienoin saavutukseni lienee korkeahko sijoittuminen The Party -tapahtumassa **Jugin** biisillä, joka äänestettiin jopa paremmaksi kuin Jugin oma entrybiisi. Feikkaamatta ei voi voittaa!

Joka tapauksessa demokompoja on nähty. Osa näistä on tosin luokkaa ”olin paikalla, mutta en läsnä”. Ennen muinoin demopartyjen aikataulut olivat näet erittäin venyvä eikä usein edes viitteellinen käsite. Syystä tai toisesta kisat aikataulutettiin usein yölle ja lipsuminen oli enemmän sääntö kuin poikkeus. Sitten vain odoteltiin, ja kompo alkoi kun alkoi.

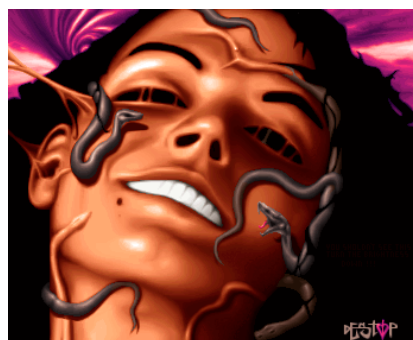
Tilannetta vaikeutti tietysti sekin, että demot ajettiin livenä. Etenkin PC-demojen kanssa tämä oli murhaa, sillä PC-rauta oli aivan tilkkutäkkisälärämää. Harrastelijoiden koodi toimi omalla koneella ja tuurilla muillakin. Kovemmat – tai kehnommat – Amiga-kooderit takoivat hekin rautaa niin tiukasti, että myös Amiga-demot saattoivat kaatuilla eri kokoonpanoissa. Edes Commodore 64 ei ollut ongelmaton.

Sitten bootattiin kone ja yritettiin ajaa demoa uudelleen, jos aikataulut ja kompocrew'n pinna vielä kestivät.

Kaikkein pahin tai mieleenpainuvien näistä katastrofaalisista demokisoista oli Assembly 1994. Demokisan alku viivästy ja viivästy niin paljon, että kaikki olivat jo aivan kuolemanväsyneitä ja henkisesti pirstaleina muutaman päivän partypuutken jälkeen. Uskollisesti kuitenkin odotimme kompon alkua – tai miten sen nyt odottamiseksi käsittää. Olimme porukalla kiivenneet jäähallin

katsomoon yläpenkeille, jossa porukka nukahteli istualleen ja lopulta kaikkensa antaneena vaipui makuulle koomaamaan ja nukkumaan äärettömän epä-mukavilla kuppipenkeillä. Aurinkokin nousi loimuttamaan kirkkaalla, pilvettömällä taivaalla ja paahtoi jäähallin lämpötilaa aina vain ylemmäs. Ilmastointi oli jo aikaa sitten nöyryntynyt tuhansien harrastajien ja tietokoneiden armadan edessä.

Mikä hienointa, aurinko helotti noustuaan hallin screeneihin niin, ettei niiden kuvaakaan enää erottanut joka puolelta. Jatkoimme sitkeästi valvontavuorottelua, ja lopulta kompot alkoivat. Mutta voih, demoja oli tietenkin monituisia ja osa niin tavanomaisia, ettei väsynyt mieli jaksanut enää keskittyä. Koko noin kymmenen skenerin porukkamme nuokkui ja nukkui kompojen läpi. Kuin ihmeen kaupalla joku onnistui silti pysyttelemään hereillä ja ravisteli muita katsomaan niitä parempia demoja tai efektejä. Mutta jos demo oli tylsä, uni tuli taas...



Grafiikkakilpailun antimia: Marble Face by Slaine/Ivory ja Pain by Destop/Cyberiad & Carillon.

Ja näin Assembly 1994:n demokisa oli minulle se demoskaba, jonka näin vain pätkittäin ja josta muistan vielä vähemmän, vaikka olin paikalla alusta loppuun. Muistikuvat eivät enää kerro, valuumeko yläpenkeiltämme jonnekin nukkumaan vai suljimme silmämme niille sijoillemme.

Ainoa selkeä muistikuva demokompoista on hetki, kun jossain demossa ruudulla pyöri söpö nalle ja **Deadbeat/The Sharks** huusi siihen ”HAZI TYKKÄÄ HINAJASTA” äänellä, joka kantoi ja kaikui koko halliin. Hazi eli **Hazard/Byterapers** tietenkin repesi räätiväsyneenä täysin – niin muutkin. Toinen muistijälki väittää, että oman C-64-demomme *World of Code II:n* latausrutiini olisi seonnut levyaseman muistiin edellisestä demosta jääneestä rutiinista. En muista, katsottiinko demo uudelleen, mutta siihenhän se C-64-kompovoitto kaatui, levyaseman aiheuttamaan bugiin. Sanoinko aiemmin jotain demojen näyttämisestä liivenä?

Ehkä siis ymmärrätte, miksi ainakin minä arvostan suuresti Assemblyn päätöstä järjestää demokompo järkevään aikaan. Jos yöllisiä kompoja ei jaksanut edes nuorena ja reippaana, niin vielä vähemmän tässä kypsässä iässä.

Enkä mene nyt edes siihen, kuinka The Partyillä nukahtelin istualleni PC-demokompossa ja kuorsasin kaverin olkapäätä vasten – tai kuinka olin baarissa juuri silloin, kun Spaceballsin *State of the Art* näytettiin ensimmäisen kerran. 🍷

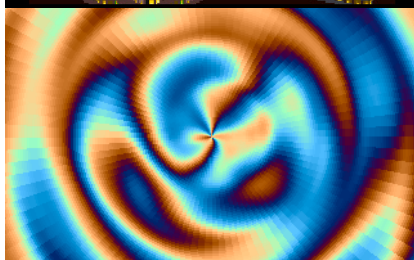
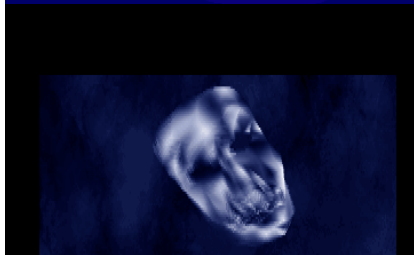
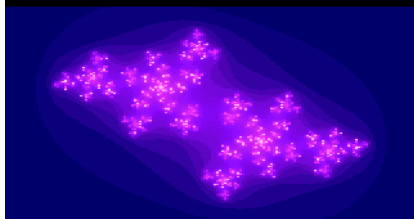
Lisätietoja Assembly 1994:stä

Tulokset: demoparty.net/assembly1994/results.html

Teokset: ftp.scene.org/pub/parties/1994/assembly94/

Asm 1994 CD-ROM: archive.org/details/ASMROM94

STELLAR



Assembly 1994:n demoja 25 vuotta myöhemmin

Amiga-demot

Despair by Insane, 7. sija
Vähän turhan mielikuvituksen efektidemo, jonka aikakautensa hittejä apinoiva biisi kuulostaa tänään kovin kaoottiselta. Lyhytkin on. **2/5**

Cccp by Juliet & Case, 5. sija.
Mainion ärhäkkä teknotemo tyylikkällä visuaaleilla ja tavanomaista poikkeellisemmalla asenteella. **3/5**

Drool This by Parallax, 3. sija
Hieno AGA-raudalle tehty demo, jossa PC-demosten vaikutus ja haaste näkyy liikaakin. Vaikka Parallaxin jampat takoivatkin uutta Amigaa rajusti, niin demo tuntuu tänään katsottuna aivan liikaa todistelulta: kyllä Amigakin tekee *Wolfenstein*-käytäviä ja 3D-maisemia! Parhaimmillaan demo on silloin, kun nykivästä 3D:stä astutaan 2D-maailmaan ja hyödynnetään Amigan omimpia kykyjä. **3/5**

Breath Taker by Virtual Dreams / Fairlight, 2. sija
VDFLT eli äärimmäisen luomisvoimaista vaihetta 1990-luvun alun ja puolivälin tienoilla. Kakkossijalle noussut demo ei kuitenkaan ole porukan parhaimmista, vaikka siinä vilahtelee aikakauden suosittuja demoeffektejä. Loppuvaikutelma on vain jotenkin kaoottinen, eikä demossa ole punaista lankaa. **3/5**

Mindflow by Stellar, 1. sija
Stellarin voittodemo alkaa aikaansa ja rautaansa nähden huikalla fraktaalimaisemoinnilla, jonka jälkeen matka jatkuu ysäriin Amiga-demoparhaimmiston lukeutuvan popituksen tahdissa päättömän hienosta efektistä ja koodivelhoudesta toiseen. Demo pysyy hyvin kasassa ja eri osioiden välillä liikutaan joko taitavien ajojen tai hyvin synkattujen feidien siivittämänä. Tosin tässäkin demossa apinoidaan paikoin liikaa PC-maailmasta. **4/5**

PC-demot

the message from the cosmic brothers by Complex, 9. sija
Mestarikaksikko Jugin ja Jmagicin videopätkistä rakennettu tunnelmointidemo on hypnoottinen kasa teko-taiteellista näkemystä ja visuaalista leikkittelyä. Teoksessa on hatunnostoa ja muun muassa 242:n ja State of the Artin suuntaan, mutta se ei yritäkään viihdyttää. Tätä ei tanssita, tätä tuijotetaan juovuksissa. **4/5**

Heartquake by Iquana, 3. sija
Turhan vaisuksi jäävä fraktaaleja ja bobeja pursuava teos vihjaa hetkitäin tiukemmastakin menosta, mutta ei kuitenkaan lunasta lupauksiaan. Nojatakko teknoon ja rave-meininkiin vai tyytykö pehmeisiin sampleihin? Matkan varrelle mahtuu oivaltaviakin ideoita ja designia, mutta näin jälkikäteen mussuttaen tiimin olisi kannattanut tarttua kunnolla omaperäisimpiin ideoihinsa ja rakentaa demo niiden ympärille. Nyt tyyllilaji ei pysy kasassa. Pisteet morfauskrediiteistä, siinä on oltu ajan hermolla. **3/5**

Holistic by Cascada VR-section, 2. sija
PC tarjosi demontekijöille rutkasti tehoa, ja Amiga-demoihin verrattuna 3D-efektit ja -objektit pyörivät ylivermaisella nopeudella. Kokonaisuus on kuitenkin tyyliä, ja demo on liimattu kasaan kehnosti, jopa tylsästi. Muutamaa koodinpätkää toistetaan pikku variaatioilla ja erilaisilla 3D-objekteilla loputtomiin. Donitsi pyörii kauniisti, mutta katsoja nukahtaa kotisohvalle. **2/5**

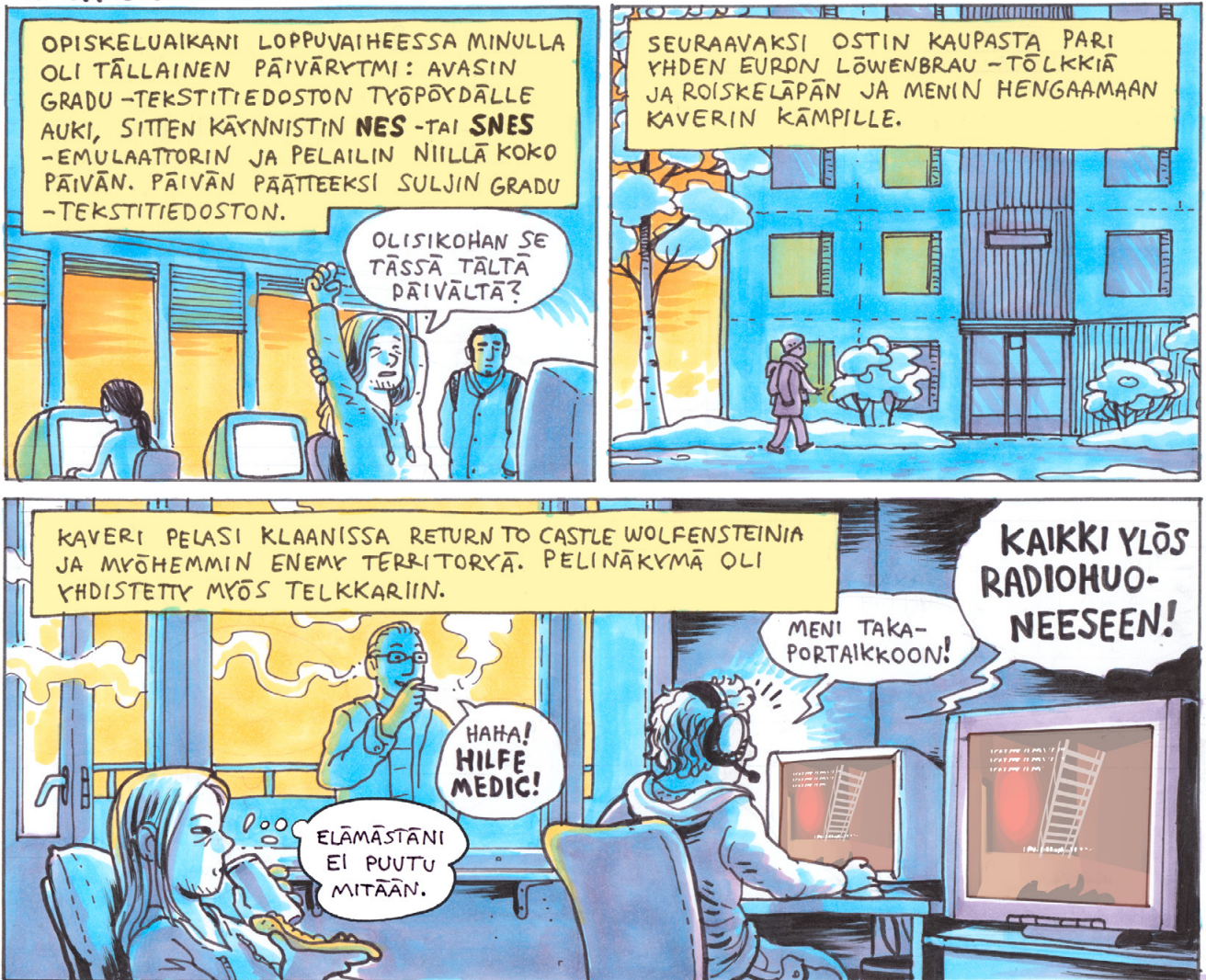
Verses by Electro Motive Force, 1. sija
Uransa toisen Assembly-demovoiton napannut EMF näyttää, kuinka ohjelmointikikat, muodikkaat efektit (mainitsiko joku jo morfauksen?), menevä tanssimusiikki ja visuaalinen design yhdistyvät toimivaksi kokonaisuudeksi. Tämän tekijöillä riitti visiota elävän, uudistuvan ja koko ajan jotain uutta tarjoavan teoksen luomiseen. **4/5**

"LET'S PLAY" VUONNA 1993

JU NÄÄS -19



E-URHEILUA VUONNA 2003



SKROLLI

TIETOKONEKULTTUURIN ERIKOISLEHTI



Ensi vuonna **Skrolli** laajenee, kun tutun sisällön lisäksi **Retro Rewind** -pelilehti alkaa ilmestyä Skrollin liitteenä. Kotiin tai konttorille – tilaa uusi tai uusi tilauksesi:

tilaaskrolli.fi

Paperilehti (40 €), digilehti (iOS/Android) tai PDF-lehti



DEVELOPER MEETUP²

Solita Core

24.10.19

KELLOHALLI, HELSINKI

250+ Software Developers · Tech
Talks · Upskilling workshops · Food
& Drinks · Chill Time & Cool People

- > JOIN US: WWW.LYYTI.IN/SOLITA-CORE-2019
- > FOLLOW OUR BLOG: DEV.SOLITA.FI